SIEMENS

SIMATIC

ET 200S
Interfacemodul IM 151-7 CPU

Handbuch

Diese Dokumentation ist Bestandteil des Dokumentationspaketes
6ES7 151-1AA00-8AA0

Ausgabe 09/2002
A5E00058779-02

Vorwort, Inhaltsverzeichnis

Produktübersicht 1

Kurzanleitung zur Inbetriebnahme (Getting Started) 2

Adressierung 3

ET 200S im PROFIBUS-Netz 4

ET 200S im MPI-Netz 5

Inbetriebnahme und Diagnose 6

Funktionen des IM 151-7 CPU 7

Zyklus- und Reaktionszeiten 8

Technische Daten 9

Umstieg von IM 151/CPU (6ES7 151-7Ax00-0AB0) auf IM 151-7 CPU (6ES7 151-7Ax10-0AB0) 10

Operationsliste A

Ausführungszeiten der SFCs und SFBs B

Stellung des IM 151-7 CPU in der CPU-Landschaft C

Glossar, Index

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährdungsgrad folgendermaßen dargestellt:

Gefahr

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Warnung

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

bedeutet, dass eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

bedeutet, dass ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Hinweis

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:

Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Warenzeichen

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind Marken der SIEMENS AG.

Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

Copyright Siemens AG 1998 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung

Siemens AG
Bereich Automation and Drives
Geschäftsgebiet Industrial Automation Systems
Postfach 4848, D-90327 Nürnberg

Haftungsausschluß

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard-und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

© Siemens AG 2002
Technische Änderungen bleiben vorbehalten.

Excellence in Automation & Drives: Siemens

Vorwort

Zweck des Handbuchs

Das vorliegende Handbuch stellt eine Ergänzung zum Handbuch *Dezentrales Peripheriesystem ET 200S* dar. Es enthält die Beschreibung sämtlicher Funktionen des Interfacemoduls IM 151-7 CPU. Das Handbuch enthält nicht Funktionen, die ET 200S generell betreffen. Diese finden Sie im Handbuch *Dezentrales Peripheriesystem ET 200S* (siehe auch Abschnitt Lieferpaket).

Die Informationen des vorliegenden Handbuchs und des Handbuchs *Dezentrales Peripheriesystem ET 200S* ermöglichen es Ihnen, ET 200S mit dem Interfacemodul IM 151-7 CPU als DP-Slave am PROFIBUS-DP oder in einem MPI-Netz zu betreiben.

Erforderliche Grundkenntnisse

Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik erforderlich.

Außerdem werden Kenntnisse über die Verwendung von Computern oder PC-ähnlichen Arbeitsmitteln (z. B. Programmiergeräten) unter dem Betriebssystem Windows 95/98/2000 bzw. NT vorausgesetzt. Sie sollten auch Kenntnisse im Umgang mit der Basissoftware STEP 7 haben. Diese werden im Handbuch “Programmieren mit STEP 7 V5.1” vermittelt.

Gültigkeitsbereich des Handbuchs

Das vorliegende Handbuch ist gültig für die Interfacemodule IM 151-7 CPU mit den Bestellnummern 6ES7 151-7AA10-0AB0 und 6ES7 151-7AB10-0AB0 und die im Handbuch *Dezentrales Peripheriesystem ET 200S* angegebenen Komponenten des Dezentralen Peripheriesystems ET 200S.

Dieses Handbuch enthält eine Beschreibung der Komponenten, die zum Zeitpunkt der Herausgabe des Handbuchs gültig sind. Wir behalten uns vor, neuen Komponenten und Komponenten mit neuem Ausgabestand eine Produktinformation mit aktuellen Informationen beizulegen.

Änderungen gegenüber der Vorgängerversion

Gegenüber der Vorgängerversion enthält das vorliegende Handbuch folgende Änderungen/Ergänzungen:

- Änderung der Modulbezeichnung in IM 151-7 CPU
- koexistente Schnittstelle MPI/DP
- neues Speicherkonzept
- zusätzliche Kommunikationsdienste
- neue Bausteine
- MMC bis 8 MByte
- Datenablage und Projektspeicherung auf MMC
- Betriebsstundenzähler 32 Bit

Hinweis: Die Vorgängerversion dieses Handbuchs “ET 200S” erkennen Sie in der Fußzeile an der Nummer A5E00058779-01.

Die jetzige Nummer ist: A5E00058779-02.

Normen und Zulassungen

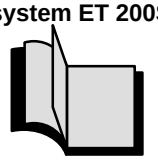
Das Dezentrale Peripheriesystem ET 200S basiert auf der Norm IEC 61784-1:2002 Ed1 CP 3/1.
Das Dezentrale Peripheriesystem ET 200S erfüllt die Anforderungen und Kriterien der IEC 61131, Teil 2 und die Anforderungen zur CE-Kennzeichnung. Für ET 200S liegen die Zulassungen für CSA, UL und FM sowie Schiffsbau vor.

Ausführliche Angaben zu den Normen und Zulassungen finden Sie im Handbuch *Dezentrales Peripheriesystem ET 200S*.

Einordnung in die Informationslandschaft

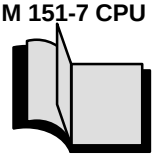
Dieses Handbuch ist Bestandteil des Dokumentationspaketes mit der Bestellnummer 6ES7 151-1AA00-8xA0 und besteht aus 6 Handbüchern mit folgenden Inhalten:

Dezentrales Peripheriesystem ET 200S



- Montieren und Verdrahten von ET 200S
- Inbetriebnahme und Diagnose von ET 200S
- Technische Daten von IM 151, digitale und analoge Elektronikmodule
- Bestellnummern für ET 200S

Interfacemodul IM 151-7 CPU



- Adressierung von IM 151-7 CPU
- ET 200S mit IM 151-7 CPU im PROFIBUS-Netz
- Inbetriebnahme und Diagnose von IM 151-7 CPU
- Technische Daten des IM 151-7 CPU
- *STEP 7*-Operationsliste

Motorstarter ET 200S

- Montieren und Verdrahten von Motorstartern
- Inbetriebnahme und Diagnose von Motorstartern
- Technische Daten von Motorstartern
- Bestellnummern für Motorstartern

Technologische Funktionen ET 200S

- 1COUNT 24V/100kHz
- 1COUNT 5V/500kHz
- 1SSI
- 2PULSE

Positionieren ET 200S

- 1STEP 5V/204kHz
- 1POS INC/DIGITAL
- 1POS SSI/DIGITAL
- 1POS INC/ANALOG
- 1POS SSI/ANALOG

Serielle Schnittstellenbaugruppe ET 200S

- 1 SI 3964/ASCII
- 1 SI MODBUS/USS

Hinweis

Das Handbuch *ET 200S Fehlersichere Module* befindet sich im Dokumentationspaket *S7 F Systems*.

Wegweiser

Um Ihnen den schnellen Zugriff auf spezielle Informationen zu erleichtern, enthält das Handbuch folgende Zugriffshilfen:

- Am Anfang des Handbuches finden Sie ein vollständiges Gesamtinhaltsverzeichnis und jeweils eine Liste der Bilder und Tabellen, die im gesamten Handbuch enthalten sind.
- In den Kapiteln finden Sie auf jeder Seite in der linken Spalte Informationen, die Ihnen einen Überblick über den Inhalt des Abschnitts geben.
- Im Anschluss an die Anhänge finden Sie ein Glossar, in welchem wichtige Fachbegriffe definiert sind, die im Handbuch verwendet wurden.
- Am Ende des Handbuchs finden Sie ein ausführliches Stichwortverzeichnis, welches Ihnen den schnellen Zugriff auf die gewünschte Information ermöglicht.

Besondere Hinweise

Zusätzlich zu den Handbüchern für ET 200S benötigen Sie das Handbuch zu dem eingesetzten DP-Master und die Dokumentation zur eingesetzten Projektier- und Programmiersoftware (Aufstellung siehe Handbuch *Dezentrales Peripheriesystem ET 200S*, Anhang A).

Hinweis

Eine Aufstellung der Handbuchinhalte der ET 200S-Handbücher finden Sie im Kapitel 1.2 des vorliegenden Handbuchs.

Wir empfehlen Ihnen, sich zuerst in diesem Kapitel zu orientieren, welche Inhalte in welchem Handbuch für die Lösung Ihrer Aufgabenstellung wichtig sind.

Recycling und Entsorgung

Das IM 151-7 CPU ist aufgrund seiner schadstoffarmen Ausrüstung recyclingfähig. Für ein umweltverträgliches Recycling und die Entsorgung Ihres Altgerätes wenden Sie sich an einen zertifizierten Entsorgungsbetrieb für Elektronikschrott.

Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

`http://www.siemens.com/automation/partner`

Trainingscenter

Um Ihnen den Einstieg in das Dezentrale Peripheriesystem ET 200S und das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in
D 90327 Nürnberg.
Telefon: +49 (911) 895-3200

Internet: `http://www.sitrain.com`

A&D Technical Support

Weltweit erreichbar zu jeder Tageszeit:

Technical Support

Weltweit (Nürnberg) **Technical Support** Ortszeit: 0:00 bis 24:00 / 365 Tage Telefon: +49 (0) 180 5050-222 Fax: +49 (0) 180 5050-223 E-Mail: adsupport@ siemens.com GMT: +1:00		
Europa / Afrika (Nürnberg) **Authorization** Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +49 (911) 895-7200 Fax: +49 (911) 895-7201 E-Mail: adsupport@ siemens.com GMT: +1:00	**Amerika** (Johnson City) **Technical Support and Authorization** Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +1 (0) 770 740 3505 Fax: +1 (0) 779 740 3699 E-Mail: isd-callcenter@ sea.siemens.com GMT: -5:00	**Asien / Australien** (Singapur) **Technical Support and Authorization** Ortszeit: Mo.-Fr. 8:30 bis 17:30 Telefon: +65 (0) 740-7000 Fax: +65 (0) 740-7001 E-Mail: simatic.hotline@ sae.siemens.com.sg GMT: +8:00
Technical Support und Authorization sprechen generell Deutsch und Englisch.		

Service & Support im Internet

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser komplettes Wissen online an.

`http://www.siemens.com/automation/service&support`

Dort finden Sie:

- den Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche in Service & Support.
- ein Forum, in welchem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Automation & Drives vor Ort über unsere Ansprechpartner-Datenbank.
- Informationen über Vor-Ort Service, Reparaturen, Ersatzteile. Vieles mehr steht für Sie unter dem Bergriff ”Leistungen” bereit.

Inhaltsverzeichnis

Vorwort

1 Produktübersicht

1.1 Was ist das Interfacemodul IM 151-7 CPU? 1-2

1.2 Wegweiser durch die ET 200S-Handbücher 1-5

2 Kurzanleitung zur Inbetriebnahme (Getting Started)

2.1 1. Schritt: Montieren von IM 151-7 CPU (ET 200S) und S7-300 2-3

2.2 2. Schritt: Verdrahten von IM 151-7 CPU (ET 200S) und S7-300 2-4

2.3 3. Schritt: In Betrieb nehmen des IM 151-7 CPU (ET 200S) 2-6

2.4 4. Schritt: Konfigurieren des IM 151-7 CPU für Stand-Alone-Betrieb (MPI) 2-7

2.5 5. Schritt: Programmieren des IM 151-7 CPU 2-9

2.6 6. Schritt: Probelauf 2-10

2.7 7. Schritt: Umrüsten des IM 151-7 CPU als DP-Slave und in Betrieb nehmen der S7-300 2-11

2.8 8. Schritt: Konfigurieren des IM 151-7 CPU als DP-Slave und der S7-300 als DP-Master 2-12

2.9 9. Schritt: Programmieren des IM 151-7 CPU und der S7-300-CPU 2-17

2.10 10. Schritt: In Betrieb nehmen und Probelauf von IM 151-7 CPU und S7-300 2-20

3 Adressierung

3.1 Steckplatzorientierte Adressierung der Peripheriemodule 3-2

3.2 Freie Adressierung der Peripheriemodule 3-4

3.3 Datenaustausch mit dem DP-Master 3-5

3.4 Zugriff auf den Übergabespeicher im IM 151-7 CPU 3-7

4 ET 200S im PROFIBUS-Netz

4.1 ET 200S im PROFIBUS-Netz 4-2

4.2 Netzkomponenten 4-6

4.3 PROFIBUS-Adresse 4-8

4.4 Funktionen über PG/OP 4-9

4.5 Direkter Datenaustausch 4-12

5 ET 200S im MPI-Netz

5.1 ET 200S im MPI-Netz 5-2

5.2 MPI-Adresse 5-3

6 Inbetriebnahme und Diagnose

6.1 IM 151-7 CPU projektieren 6-2

6.2 IM 151-7 CPU urlöschen 6-4

6.3 Inbetriebnahme und Anlauf von ET 200S 6-7

6.4 Diagnose durch LED-Anzeige 6-9

6.5 Diagnose über Diagnoseadresse mit STEP 7 6-11

6.6 Slave-Diagnose bei Einsatz des IM 151-7 CPU als I-Slave 6-14
6.6.1 Stationsstatus 1 bis 3 6-15
6.6.2 Master-PROFIBUS-Adresse 6-17
6.6.3 Herstellerkennung 6-18
6.6.4 Kennungsbezogene Diagnose 6-19
6.6.5 Modulstatus 6-20
6.6.6 Alarmstatus 6-22

6.7 Diagnosedaten der Elektronikmodule 6-25
6.7.1 Diagnosedaten der Elektronikmodule im Anwenderprogramm auswerten 6-25
6.7.2 Aufbau und Inhalt der Diagnosedaten Bytes 0 bis 7 6-27
6.7.3 Kanalspezifische Diagnosedaten ab Byte 8 6-29
6.7.4 Beispiel: ET 200S-Modul: 2 AI U (6ES7 134-4FB00-0AB0) mit je einer Diagnose für Kanal 0 und 1 6-30

7 Funktionen des IM 151-7 CPU

7.1 Daten für PROFIBUS-DP 7-2

7.2 Betriebsartenschalter und Anzeigeelemente 7-4

7.3 SIMATIC Micro Memory Card 7-6

7.4 Speicherkonzept 7-12
7.4.1 Speicherbereiche des IM 151-7 CPU 7-12
7.4.2 Speicherfunktionen 7-15
7.4.3 Operandenbereiche 7-20
7.4.4 Handling von Daten in DB 7-23
7.4.5 Speichern/Holen ganzer Projekte auf/von Micro Memory Card 7-27

7.5 Schnittstelle 7-29

7.6 Uhr 7-31

7.7 Kommunikation 7-32

7.8 Bausteine 7-36

7.9 Parameter 7-39

7.10 Parametrierung der Vergleichsstelle beim Anschluss von Thermoelementen 7-41

7.11 Ziehen und Stecken von Modulen im laufenden Betrieb 7-43

7.12 Aus- und Einschalten von Powermodulen im laufenden Betrieb 7-46

8 Zyklus- und Reaktionszeiten

8.1 Zykluszeit 8-2
8.2 Reaktionszeit 8-5
8.3 Alarmreaktionszeit 8-8

9 Technische Daten

9.1 Technische Daten des IM 151-7 CPU 9-2

10 Umstieg von IM 151-7 CPU (6ES7 151-7Ax00-0AB0) auf IM 151-7 CPU (6ES7 151-7Ax10-0AB0)

A Operationsliste

A.1 Operanden und Parameterbereiche A-2
A.2 Abkürzungen A-3
A.3 Register A-3
A.4 Adressierungsbeispiele A-5
A.5 Ausführungszeiten bei indirekter Adressierung A-7
A.5.1 Beispiel für speicherindirekte, bereichsinterne Adressierung A-9
A.5.2 Beispiel für registerindirekte, bereichsinterne Adressierung A-10
A.5.3 Beispiel für registerindirekte, bereichsübergreifende Adressierung A-11
A.5.4 Beispiel für Adressierung über Parameter A-12
A.6 Verknüpfungsoperationen mit Bitoperanden A-13
A.7 Verknüpfungsoperationen von Klammerausdrücken A-15
A.8 ODER-Verknüpfung von UND-Funktionen A-16
A.9 Verknüpfungsoperationen mit Zeiten und Zählern A-17
A.10 Verknüpfungsoperationen mit dem Inhalt von AKKU1 . . . . . A-18
A.11 Verknüpfungsoperationen mit Anzeigenbits A-19
A.12 Flankenoperationen A-21
A.13 Setzen/Rücksetzen von Bitoperanden A-22
A.14 VKE direkt beeinflussende Operationen A-23
A.15 Zeitoperationen A-24
A.16 Zähloperationen A-25
A.17 Ladeoperationen A-26
A.18 Ladeoperationen für Timer und Zähler A-28
A.19 Transferoperationen A-29
A.20 Lade- und Transferoperationen für Adressregister A-31
A.21 Lade- und Transferoperationen für das Statuswort A-32
A.22 Ladeoperationen für DB-Nummer und DB-Länge A-32
A.23 Festpunktarithmetik (16-Bit) A-33
A.24 Festpunktarithmetik (32 Bit) A-34

A.25 Gleitpunktarithmetik (32 Bit) A-35
A.26 Addition von Konstanten A-35
A.27 Addition über Adressregister A-36
A.28 Vergleichsoperationen mit Ganzzahlen (16 Bit) A-36
A.29 Vergleichsoperationen mit Ganzzahl (32 Bit) A-37
A.30 Vergleichsoperationen (32-Bit-Realzahlen) A-38
A.31 Schiebeoperationen A-39
A.32 Rotieroperationen A-40
A.33 AKKU-Transferoperationen, Inkrementieren, Dekrementieren A-41
A.34 Bildoperation, Nulloperation A-41
A.35 Datentyp-Umwandlungsoperationen A-42
A.36 Komplementbildung A-43
A.37 Baustein-Aufrufoperationen A-44
A.38 Baustein-Endeoperationen A-45
A.39 Tausche Datenbausteine A-45
A.40 Sprungoperationen A-46
A.41 Operationen für das Master Control Relay (MCR) A-48
A.42 Systemzustandsliste (SZL) A-49

B Ausführungszeiten der SFCs und SFBs

B.1 Systemfunktionen (SFCs) B-1
B.2 Systemfunktionsbausteine (SFBs) B-4

C Stellung des IM 151-7 CPU in der CPU-Landschaft

C.1 Unterschiede zu ausgewählten S7-300 CPUs C-2
C.2 Portierung des Anwenderprogrammes C-3

Glossar

Index

Tabellen

1-1	Themen der Handbücher des Handbuch-Pakets ET 200S	1-6
3-1	Adressen der Peripheriemodule von ET 200S	3-3
3-2	Zugriffe auf die Adressbereiche	3-7
3-3	Adressieroberfläche in STEP 7 V5.1 (Ausschnitt)	3-8
4-1	Verhalten des IM 151-7 CPU in Abhängigkeit von der Einstellung der DP-Schnittstelle	4-4
4-2	Netzkomponenten	4-6
6-1	Projektiermöglichkeiten	6-2
6-2	Möglichkeiten zum Urlöschen	6-4
6-3	CPU-interne Vorgänge beim Urlöschen	6-5
6-4	LED-Anzeige für PROFIBUS-DP	6-10
6-5	Reaktionen auf Betriebszustandsänderungen bzw. Unterbrechungen des Nutzdatentransfers im DP-Master und im ET 200S mit IM 151-7 CPU	6-12
6-6	Auswertung von RUN-STOP-Übergängen im DP-Master/im ET 200S	6-13
6-7	Aufbau von Stationsstatus 1 (Byte 0)	6-15
6-8	Aufbau von Stationsstatus 2 (Byte 1)	6-16
6-9	Aufbau von Stationsstatus 3 (Byte 2)	6-16
6-10	Aufbau der Master-PROFIBUS-Adresse (Byte 3)	6-17
6-11	Aufbau der Herstellerkennung (Byte 4, 5)	6-18
6-12	Kennungen der Modulklassen	6-27
7-1	Eigenschaften aus der GSD-Datei	7-2
7-2	Stellungen des Betriebsartenschalters	7-4
7-3	LED-Anzeige für CPU-Funktionalität	7-5
7-4	Verfügbare MMCs	7-7
7-5	Firmware-Update mit MMC	7-10
7-6	Sichern der Betriebssystems	7-11
7-7	Remanenzverhalten der Speicherobjekte	7-14
7-8	Operandenbereiche des Systemspeichers	7-20
7-9	Anschließbare Geräte	7-30
7-10	Eigenschaften der Uhr	7-31
7-11	Kommunikationsdienste des IM 151-7 CPU	7-32
7-12	GD-Ressourcen des IM 151-7 CPU	7-35
7-13	Übersicht: Bausteine	7-36
7-14	OBs für Zyklus und Anlauf	7-37
7-15	OBs fürAlarme	7-37
7-16	OBs für Fehlerreaktionen	7-38
7-17	Parameterblöcke, einstellbare Parameter und ihre Wertebereiche für IM 151-7 CPU	7-39
7-18	Parametrierung der Vergleichsstelle	7-41
7-19	Ergebnis des Soll-Ist-Vergleiches bei nicht parametrierbaren Modulen	7-44
7-20	Ergebnis des Soll-Ist-Vergleiches bei parametrierbaren Modulen, Powermodul eingeschaltet	7-44
7-21	Ergebnis des Soll-Ist-Vergleiches bei parametrierbaren Modulen, Powermodul ausgeschaltet	7-45
8-1	Betriebssystembearbeitungszeit im Zykluskontrollpunkt	8-3
8-2	Prozessabbild-Aktualisierung	8-3
8-3	Abhängigkeit der Anwenderprogrammbearbeitungszeit	8-4
8-4	Zyklusverlängerung durch Einschachteln von Alamen	8-4
8-5	Alarmreaktionszeiten des IM 151-7 CPU (ohne Kommunikation)	8-8
9-1	Anschlussbelegung des Interfacemoduls IM 151-7 CPU	9-3
A-1	SZL-Teillisten des IM151/CPU	A-49

C-1 Unterschiede zu ausgewählten S7-300 CPUs C-2
C-2 Beispiel: Ersetzungen unter Extra → Umverdrahten C-4

Bilder

1-1 Ansicht des Dezentralen Peripheriesystems ET 200S mit IM 151-7 CPU 1-3
1-2 Komponenten und benötigte Handbücher 1-5
2-1 Montieren von IM 151-7 CPU (ET 200S) 2-3
2-2 Ansicht der S7-300 2-5
3-1 Aufbau des defaultmäßig eingestellten Adressraums 3-2
3-2 Steckplätze bei ET 200S 3-2
3-3 Beispiel für die Adresszuordnung zu Peripheriemodulen 3-3
3-4 Aufbau des Adressraums bei freier Adressierung 3-4
3-5 Prinzip des Datenaustauschs zwischen DP-Master und ET 200S mit IM 151-7 CPU 3-5
4-1 Beispiel für ein PROFIBUS-Netz 4-2
4-2 Betriebsarteneinstellung der DP-Schnittstelle am IM 151-7 CPU 4-3
4-3 PG/OP greift über DP-Schnittstelle im DP-Master auf ET 200S zu 4-5
4-4 PG greift direkt auf ET 200S zu 4-5
4-5 DP-Netz anschließen 4-7
4-6 Prinzip des Forcens 4-10
4-7 Direkter Datenaustausch mit IM 151-7 CPU 4-12
5-1 Beispiel für ein MPI-Netz 5-2
6-1 Bedienfolge des Betriebsartenschalters zum Urlöschen 6-5
6-2 Diagnoseadressen für DP-Master und ET 200S 6-11
6-3 Aufbau der Slave-Diagnose 6-14
6-4 Aufbau der kennungsbezogenen Diagnose des IM 151-7 CPU 6-19
6-5 Aufbau des Modulstatus 6-21
6-6 Aufbau des Alarmstatus 6-22
6-7 Byte y+4 bis y+7 für Diagnosealarm (Betriebszustandswechsel des I-Slave) 6-23
6-8 Byte y+4 bis y+7 für Diagnosealarm (SFB 75) 6-24
6-9 Aufbau der Diagnosedaten am Beispiel eines 4 kanaligen Mischmoduls 6-26
6-10 Bytes 0 und 1 der Diagnosedaten 6-27
6-11 Bytes 4 bis 7 der Diagnosedaten 6-28
6-12 Einzelfehler eines Kanales 6-29
7-1 Betriebsartenschalter 7-4
7-2 Position des Modulschachtes für die MMC-Karte am IM 151-7 CPU 7-9
7-3 Speicherbereiche eines IM 151-7 CPU 7-12
7-4 Lade- und Arbeitsspeicher 7-15
7-5 Bearbeitungsschritte innerhalb eines Zyklus 7-21
7-6 Handling von Rezepturdaten 7-23
7-7 Handling von Messwertarchiven 7-25
7-8 Beispiel für ein Parametrierfenster der CPU-Baugruppendaten in STEP 7 V5.1 + SP4 7-42
8-1 Teile der Zykluszeit 8-2
8-2 Kürzeste Reaktionszeit 8-6
8-3 Längste Reaktionszeit 8-7
9-1 Prinzipschaltbild IM 151-7 CPU 9-4
9-2 Prinzipschaltbild IM 151-7 CPU FO 9-4
C-1 Beispiel: FB mit ungepackten Adressen C-3
C-2 Beispiel: FB mit gepackten Adressen C-4
C-3 Beispiel: Umverdrahtung der Signale C-5

Produktübersicht

1

In diesem Kapitel

Die Produktübersicht informiert Sie darüber:

- wie das Interfacemodul IM 151-7 CPU im Dezentralen Peripheriesystem ET 200S einzuordnen ist und
- in welchem Handbuch des Handbuchpakets für ET 200S Sie welche Informationen finden.

Kapitelübersicht

Kapitel	Thema	Seite
1.1	Was ist das Interfacemodul IM 151-7 CPU?	1-2
1.2	Wegweiser durch die ET 200S-Handbücher	1-5

1.1 Was ist das Interfacemodul IM 151-7 CPU?

Was ist das IM 151-7 CPU?

Das IM 151-7 CPU ist eine Komponente des Dezentralen Peripheriesystems ET 200S in der Schutzart IP 20. Das Interfacemodul IM 151-7 CPU ist eine “intelligente Vorverarbeitungseinheit” (I-Slave). Damit ist es möglich, Steuerungsaufgaben zu dezentralisieren.

Ein ET 200S mit IM 151-7 CPU kann deshalb vollständig und bei Bedarf auch eigenständig eine technologische Funktionseinheit steuern und als Stand-alone-CPU eingesetzt werden. Der Einsatz des IM 151-7 CPU führt zu weiterer Modularisierung und Standardisierung von technologischen Funktionseinheiten und einfachen und übersichtlichen Maschinenkonzepten.

Wie wird das IM 151-7 CPU in ET 200S integriert?

Das Interfacemodul IM 151-7 CPU wird wie jedes andere Modul in ET 200S integriert; d. h., gleiches Aufbaukonzept, gleiche Montage und Erweiterbarkeit.

Ansicht

Im folgenden Bild finden Sie eine Beispielkonfiguration eines ET 200S mit IM 151-7 CPU.

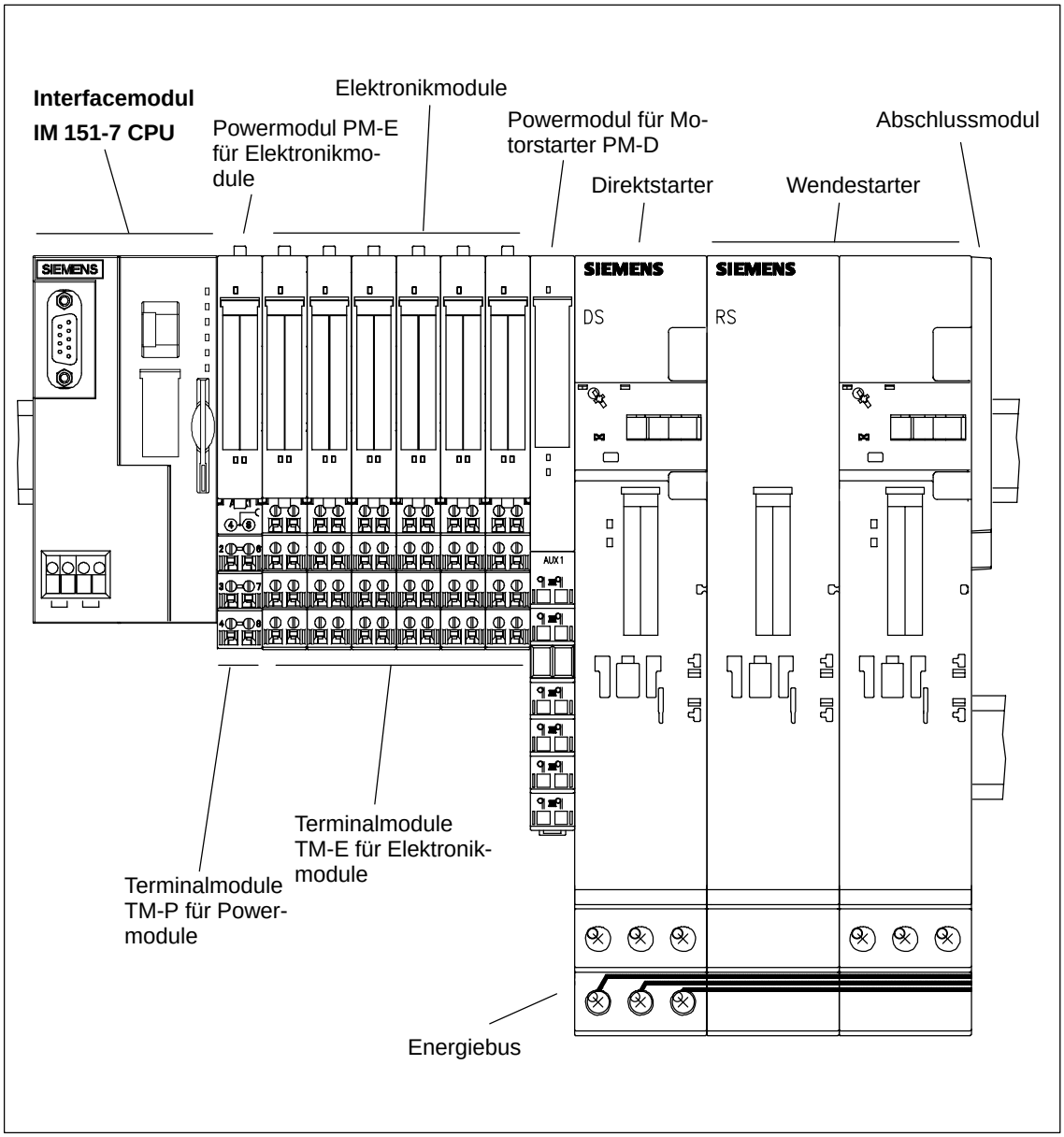


Bild 1-1 Ansicht des Dezentralen Peripheriesystems ET 200S mit IM 151-7 CPU

Eigenschaften von IM 151-7 CPU im Unterschied zu anderen Modulen

Das Interfacemodul IM 151-7 CPU verfügt über folgende, spezielle Eigenschaften:

- Das Interfacemodul verfügt über SPS-Funktionalität (integrierter CPU-Teil mit 48 kByte Arbeitsspeicher).
- Das Interfacemodul kann nur mit gestecktem Ladespeicher (MMC) betrieben werden.
- Das Interfacemodul kann mit bis zu 63 Peripheriemodulen des ET 200S-Spektrums erweitert werden.
- Das Interfacemodul verfügt über einen Betriebsartenschalter mit den möglichen Stellungen RUN, STOP und MRES.
- Es gibt 6 LEDs auf der Frontseite des Interfacemoduls zur Anzeige von:
  - Fehlern eines ET 200S (SF),
  - Busfehlern (BF),
  - Versorgungsspannung für Elektronik (ON),
  - Forceaufträgen (FRCE),
  - Betriebsart des IM 151-7 CPU (RUN und STOP).
- Varianten für den Anschluss an den PROFIBUS-DP über RS 485 und Lichtwellenleiter LWL (Variante FO)
- Die Variante FO hat zusätzlich zwei LEDs zur Anzeige von Störungen in der LWL-Übertragung (FO1F, FO2F).

Wie wird ET 200S mit IM 151-7 CPU projektiert?

Für die Projektierung von ET 200S mit IM 151-7 CPU (Konfigurierung und Parametrierung) benötigen Sie *HW Konfig* der Projektiersoftware *STEP 7*, ab Version V 5.1 + Service Pack 4. Die Vorgehensweise zur Projektierung des ET 200S mit IM 151-7 CPU finden Sie im Kapitel 6.1 des vorliegenden Handbuchs.

Wie wird IM 151-7 CPU programmiert?

Für die Programmierung des IM 151-7 CPU benötigen Sie die Projektiersoftware *STEP 7*, ab Version V 5.1 + Service Pack 4. Im Anhang A finden Sie den *STEP 7*-Operationsvorrat für die Programmierung des IM 151-7 CPU.

1.2 Wegweiser durch die ET 200S-Handbücher

Sie setzen folgende Komponenten ein ...

Die Komponenten von ET 200S sind in verschiedenen Handbüchern des Handbuch-Pakets für ET 200S beschrieben. Im folgenden Bild finden Sie mögliche Aufbau-Varianten von ET 200S und die dazu notwendigen Handbücher des Handbuch-Pakets.

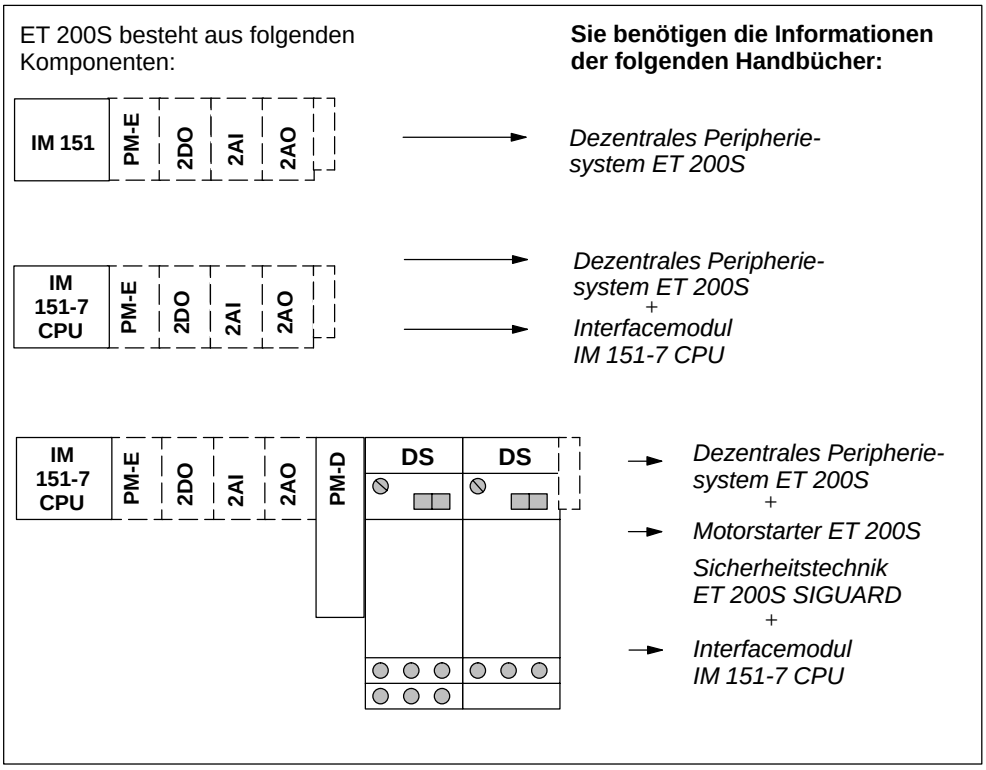


Bild 1-2 Komponenten und benötigte Handbücher

Wo finden Sie welche Information?

Die folgende Tabelle soll Ihnen als Orientierungshilfe zum schnellen Auffinden benötigter Informationen dienen. Sie erfahren, in welchem Handbuch Sie nachsehen müssen und welches Kapitel das Thema behandelt.

Tabelle 1-1 Themen der Handbücher des Handbuch-Pakets ET 200S

Thema	Handbuch			Kapitel/ Anhang
	Dezentrales Peripheriesystem ET 200S	***Interface-modul IM 151-7 CPU***	***Motor-starter ET 200S***	
Komponenten von ET 200S	x			1.2
Komponenten von ET 200S Motorstartern			x	1
Konfigurationsmöglichkeiten von ET 200S	x			3
Konfigurationsmöglichkeiten von ET 200S Motorstartern			x	1
Montieren von ET 200S; PROFIBUS-Adresse einstellen;	x			4
Montieren von ET 200S Motorstartern			x	2
Adressierung von IM 151-7 CPU		x		3
Elektrischer Aufbau und Verdrahtung von ET 200S	x			5
ET 200S mit IM 151-7 CPU am PROFIBUS-Netz		x		4
Inbetriebnahme und Diagnose von ET 200S	x			6
Inbetriebnahme und Diagnose von ET 200S mit Motorstartern			x	3
Inbetriebnahme und Diagnose von ET 200S mit IM 151-7 CPU		x		6
Funktionen des IM 151-7 CPU		x		7
Allgemeine technische Daten von ET 200S (Normen, Zulassungen, EMV, Umgebungs-bedingungen etc.)	x			7
Technische Daten von Interfacemodulen, Terminalmodulen, Power- und Elektronikmodulen	x			8, 9, 10, 11, 12
Technische Daten von ET 200S Motorstarter			x	4
Technische Daten von IM 151-7 CPU		x		9
Sicherheitstechnik ET 200S SIGUARD			x	9
Bestellnummern von ET 200S	x			A
Bestellnummern der ET 200S Motorstarter			x	A
Zyklus- und Reaktionszeit IM 151-7 CPU		x		8
Kompatibilität		x		10
STEP 7-Operationsliste		x		A
Ausführungszeiten von SFCs		x		B
Stellung des IM 151-7 CPU in der CPU-Landschaft				C
Glossar	x	x	x	Glossar

Konfigurier- und Parametriertelegramm für IM 151-7 CPU siehe im Internet unter http://www.ad.siemens.de/simatic-cs

Kurzanleitung zur Inbetriebnahme (Getting Started) 2

Einleitung

Diese Anleitung führt Sie an einem konkreten Beispiel durch 10 Inbetriebnahme-Schritte bis zu einer funktionierenden Anwendung eines IM 151-7 CPU. Dabei lernen Sie die Grundfunktion Ihres IM 151-7 CPU

- in Hardware und Software,
- im Stand-alone-Betrieb (MPI),
- als intelligenter DP-Slave (PROFIBUS-DP) kennen.

Voraussetzungen

Sie müssen grundlegend mit der Elektronik/Elektrotechnik vertraut sein und Erfahrung im Umgang mit Computer und Microsoft® Windows™ 95/98/NT/2000 besitzen.

Gefahr

Das IM 151-7 CPU, die ET 200S und die S7-300 als Bestandteil von Anlagen bzw. Systemen erfordern je nach Einsatzgebiet die Beachtung spezieller Regeln und Vorschriften.
Beachten Sie die geltenden Sicherheits- und Unfallverhütungsvorschriften, z. B. IEC 204 (NOT-AUS-Einrichtungen).
Bei Nichtbeachtung dieser Vorschriften kann es zu schweren Körperverletzungen und zur Beschädigung von Maschinen und Einrichtungen kommen.

Kapitelübersicht

Kapitel	Thema	Seite
2.1	1. Schritt: Montieren von IM 151-7 CPU (ET 200S) und S7-300	2-3
2.2	2. Schritt: Verdrahten von IM 151-7 CPU (ET 200S) und S7-300	2-4
2.3	3. Schritt: In Betrieb nehmen des IM 151-7 CPU (ET 200S)	2-6
2.4	4. Schritt: Konfigurieren des IM 151-7 CPU für Stand-Alone-Betrieb (MPI)	2-7
2.5	5. Schritt: Programmieren des IM 151-7 CPU	2-9
2.6	6. Schritt: Probelauf	2-10

Kapitel	Thema	Seite
2.7	7. Schritt: Umrüsten des IM 151-7 CPU als DP-Slave und in Betrieb nehmen der S7-300	2-11
2.8	8. Schritt: Konfigurieren des IM 151-7 CPU als DP-Slave und der S7-300 als DP-Master	2-12
2.9	9. Schritt: Programmieren des IM 151-7 CPU und der S7-300-CPU	2-17
2.10	10. Schritt: In Betrieb nehmen und Probelauf von IM 151-7 CPU und S7-300	2-20

Benötigtes Material und Werkzeug

Menge	Artikel	Bestellnummer (SIEMENS)
1	S7-300-System, bestehend aus Stromversorgung (PS), CPU mit DP-Schnittstelle (hier: CPU 315 2-DP), digitaler Eingangsbaugruppe (DI) auf Steckplatz 4 und digitaler Ausgangsbaugruppe (DO) auf Steckplatz 5, inkl. Profilschiene, Busverbinder und Verkabelung	diverse
1	Stromversorgung (PS) z. B.: PS 307 mit Netzanschlussleitung (optional)	z. B.: 6ES7 307-1EA00-0AA0
1	IM 151-7 CPU mit Abschlussmodul	z. B.: 6ES7 151-7AA10-0AB0
1	SIAMTIC Micro Memory Card (MMC)	z. B.: 6ES7 953-8LL00-0AA0
1	Powermodul (PM)	z. B.: 6ES7 138-4CA00-0AA0
1	Digitales Eingabemodul (DI)	z. B.: 6ES7 131-4BD00-0AA0
1	Digitales Ausgabemodul (DO)	z. B.: 6ES7 132-4BD00-0AA0
1	Terminalmodul (TM) für das PM	z. B.: 6ES7 193-4CC30-0AA0
2	Terminalmodule für DI und DO	z. B.: 6ES7 193-4CB30-0AA0
1	Profilschiene für ET 200S	diverse
1	Programmiergerät (PG) mit PROFIBUS-DP-Schnittstelle, installierter Software STEP 7 Version $\geq$ 5.1 und PG-Kabel (bis 1,5 MBit/s)	diverse
1	PROFIBUS-DP-Kabel	diverse
1	Schraubendreher mit Klingenbreite 3 mm	handelsüblich
1	Schraubendreher mit Klingenbreite 4,5 mm	handelsüblich
1	Seitenschneider und Werkzeug zum Abisolieren	handelsüblich
1	Werkzeug zum Aufpressen von Aderendhülsen	handelsüblich
ca. 2 m	Litze mit 1 mm^2 Querschnitt mit passenden Aderendhülsen, Form A, Länge 6 mm und 12 mm	handelsüblich
4	1-polige Ein-Taster (24 V)	handelsüblich

2.1 1. Schritt: Montieren von IM 151-7 CPU (ET 200S) und S7-300

Abfolge	Beschreibung
1	Montieren Sie die S7-300 so wie im *Installationshandbuch Automatisierungssystem S7-300, Aufbauen* beschrieben.
2	Wenn Sie das IM 151-7 CPU mit einer eigenen Stromversorgung betreiben wollen, hängen Sie die PS in die Profilschiene der S7-300 und schwenken sie ein, bis sie einrastet.
3	Hängen Sie das IM 151-7 CPU in die Profilschiene und schwenken es ein, bis es einrastet.
4	Hängen Sie das TM für das PM rechts vom IM 151-7 CPU in die Profilschiene und schwenken es ein, bis es einrastet.
5	Verschieben Sie das TM so weit nach links, bis es hörbar am IM 151-7 CPU einrastet.
6	Verfahren Sie sinngemäß zu den Punkten 3 und 4 mit zwei TM für Elektronikmodule und abschließend mit dem Abschlussmodul (rastet nicht auf der Profilschiene ein).
7	Schieben Sie das PM in das entsprechende TM, bis es einrastet.
8	Schieben Sie das DI in das linke noch freie TM, bis es einrastet.
9	Schieben Sie das DO in das letze noch freie TM, bis es einrastet.
10	Stecken Sie die Micro Memory Card in das IM 151-7 CPU, sie ist für den Betrieb zwingend erforderlich. Eine Micro Memory Card mit unbekanntem Inhalt sollte vorher am Programmiergerät gelöscht werden.

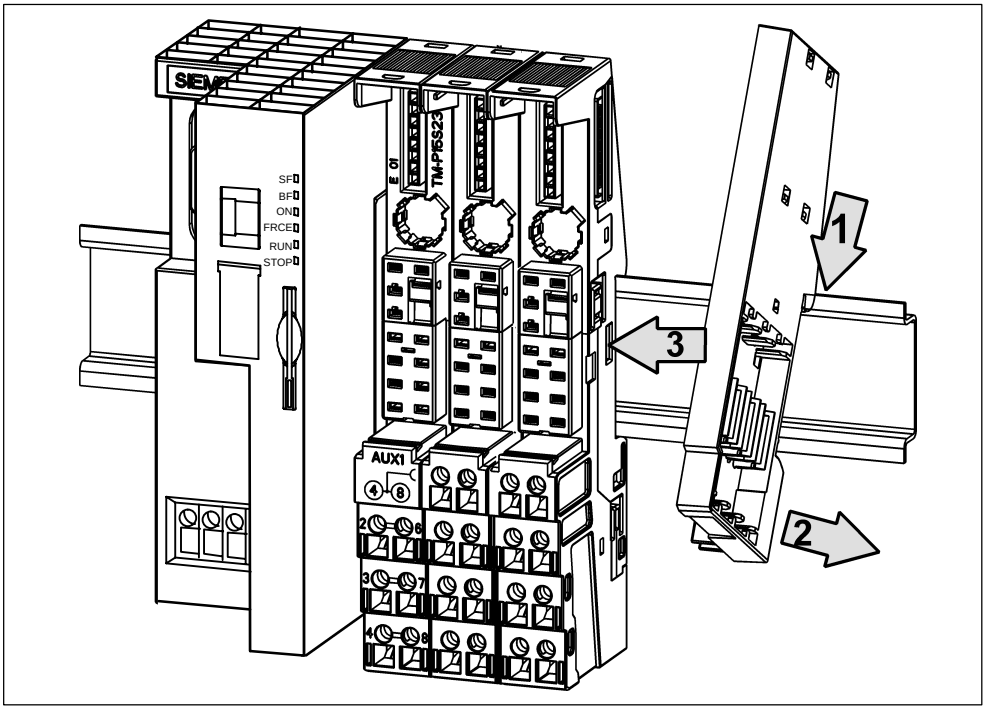


Bild 2-1 Montieren von IM 151-7 CPU (ET 200S)

2.2 2. Schritt: Verdrahten von IM 151-7 CPU (ET 200S) und S7-300

Abfolge	Beschreibung
1	Verdrahten Sie die S7-300 so wie im *Installationshandbuch Automatisierungssystem S7-300, Aufbauen* beschrieben.
2	Verlängern Sie die Anschlüsse der 4 Taster mit je einem Kabel. Isolieren Sie die freien Kabelenden auf einer Länge von 6 mm ab und verpressen Sie sie mit Aderendhülsen.
3	Verbinden Sie an der DI der S7-300 die Eingänge 1.1 (Klemme 13) und 1.2 (Klemme 14) über je einen Taster mit L+ der PS der S7-300.
4	Schließen Sie die beiden restlichen 1poligen Taster an das DI der ET 200S wie folgt an: • einen Taster an die Klemmen 1 und 3 • den anderen Taster an die Klemmen 5 und 7 **Hinweis zu Federklemmen** Lösen der Feder eines Anschlusses: Schraubendreher mit 3 mm Klingenbreite in das obere runde Loch einer Klemme bis zum Anschlag stecken und ggf. Schraubenziehergriff leicht nach oben ziehen. Danach kann ein freies Kabelende in das darunterliegende quadratische Loch eingeführt werden. Schraubendreher wieder herausziehen und den festen Sitz des Kabels überprüfen.
5	Verdrahten Sie die Klemme 2 des TMs des PM mit L+ der PS und die Klemme 3 des TMs des PM mit M der PS. Die anzuschließenden Kabelenden sind hierbei auf einer Länge von 11 mm abzuisolieren und mit Aderendhülsen zu verpressen.
6	Verdrahten Sie die Klemme 1L+ des IM 151-7 CPU mit L+ der PS und die Klemme 1M des IM 151-7 CPU mit M der PS. **Hinweis** • Die anzuschließenden Kabelenden sind hierbei auf einer Länge von 11 mm abzuisolieren und mit Aderendhülsen zu verpressen. • Sie können zur Stromversorgung des IM 151-7 CPU und des PM auch die PS der S7-300 nutzen.
7	Verbinden Sie PG und IM 151-7 CPU mit dem PG-Kabel und schrauben Sie alle Stecker fest.
8	Schließen Sie die PS der ET 200S, die PS der S7-300 und das PG an das Netz an.

Ansicht der S7-300 (nicht dargestellt ist die Verdrahtung der Spannungsversorgung der DI und der DO; PG ist an S7-300 angeschlossen)

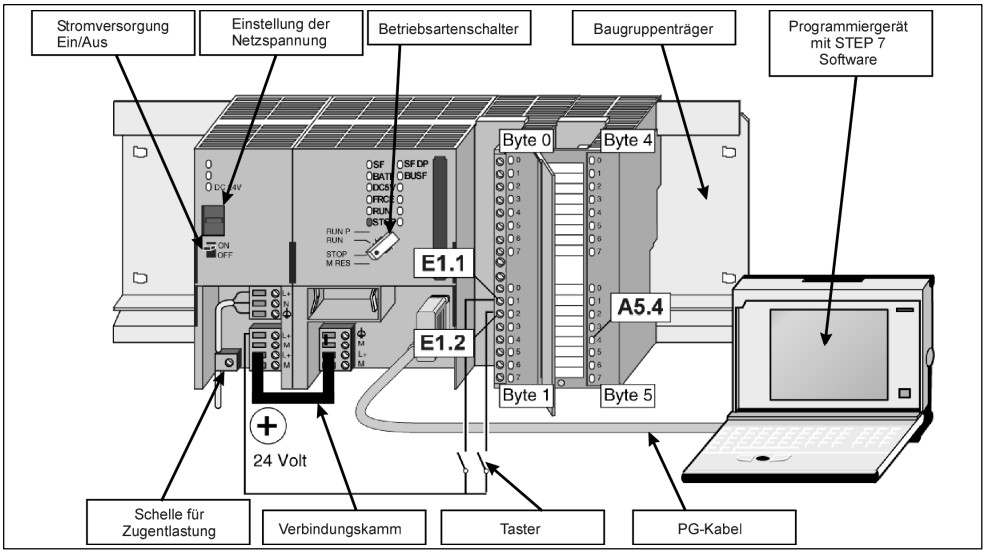
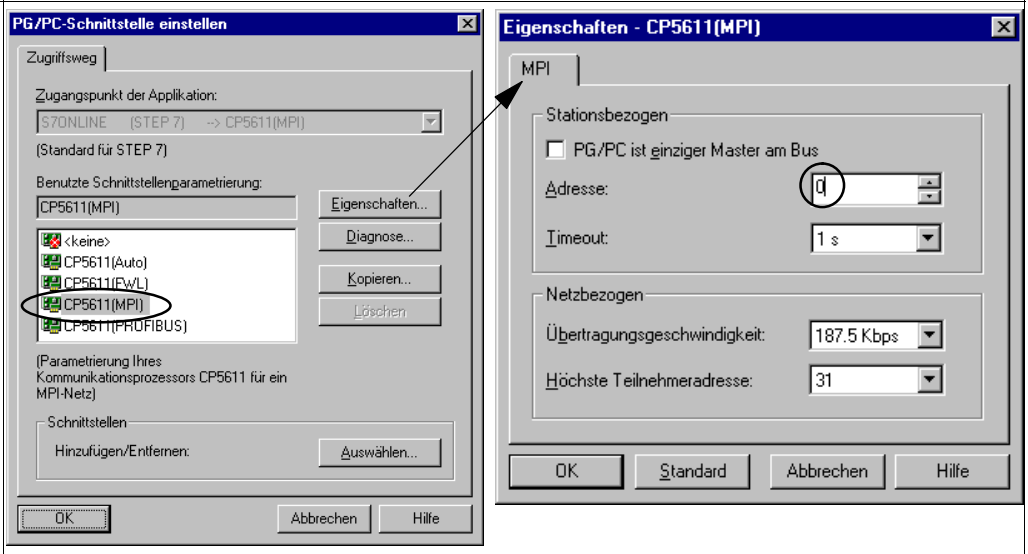


Bild 2-2 Ansicht der S7-300

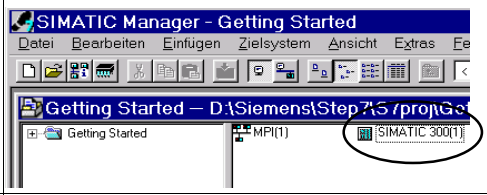
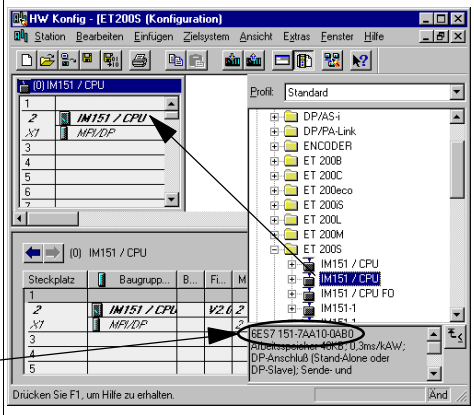
2.3 3. Schritt: In Betrieb nehmen des IM 151-7 CPU (ET 200S)

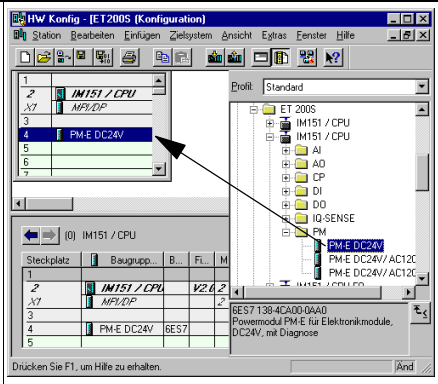
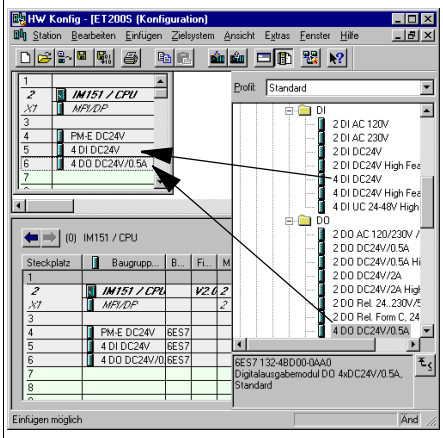
Hinweis

Bei der **Erstinbetriebnahme** (Auslieferzustand) der ET 200S ist die CPU über MPI-Adresse 2, HSA 31 und 187,5 kBaud zu erreichen.

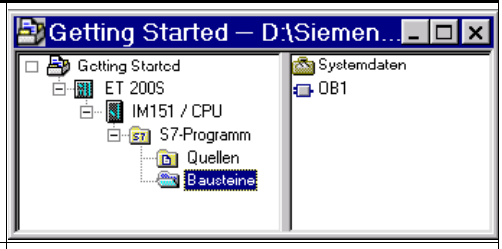
Abfolge	Beschreibung
1	Schalten Sie die PS des IM 151-7 CPU ein. **Ergebnis:** • An der PS leuchtet die *DC24V*-LED. • Am PM leuchten die *PWR*- und die *SF*-LED. • Am IM 151-7 CPU leuchten alle LED auf, die *SF-*, *BF-*, *FRCE*- und die *RUN*-LED erlöschen wieder und die *STOP*-LED beginnt schnell zu blinken. Das IM 151-7 CPU führt damit Urlöschen durch.
2	Betätigen Sie nun die beiden Taster, die am DI-Modul angeschlossen sind. Bei Betätigung des Tasters an den Klemmen 1 und 3 leuchtet die *1*-LED. Bei Betätigung des Tasters an den Klemmen 5 und 7 leuchtet die *5*-LED.
3	Schalten Sie Ihr PG ein und starten Sie auf dem Windows-Desktop den SIMATIC-Manager.
4a	Klicken Sie im SIMATIC-Manager-Hauptmenü auf Extras und wählen Sie den Menüpunkt **PG/PC-Schnittstelle einstellen**. Konfigurieren Sie die PG/PC-Schnittstelle wie folgt:
4b	 **Anmerkung:** Der Kommunikationsprozessor kann bei Ihrem PG eine andere Bezeichnung tragen. Wichtig ist, dass die MPI-Version eingestellt ist.
5	Bestätigen Sie die Einstellungen mit OK und schließen Sie das Programm PG/PC-Schnittstelle einstellen.

2.4 4. Schritt: Konfigurieren des IM 151-7 CPU für Stand-Alone-Betrieb (MPI)

Abfolge	Tätigkeit	Ergebnis
1	Erscheint im SIMATIC-Manager der Assistent für ein neues Projekt?	**Wenn ja:** Assistenten schließen, da durch den Projektassistenten das IM 151-7 CPU nicht unterstützt wird. **Wenn nein:** weiter bei Punkt 2
2	Navigieren Sie im Hauptmenü des SIMATIC-Managers zu **Datei** und wählen Sie den Menüpunkt **Neu.** Geben Sie als Projektnamen “Getting Started” an und klicken Sie auf die Schaltfläche **OK**.	Ein neues Projekt wird angelegt und geöffnet.
3	Navigieren Sie zu **Einfügen** und wählen Sie den Menüpunkt **Station.** Klicken Sie in der Liste auf: **SIMATIC 300-Station**.	
4	Benennen Sie diese Station in “ET 200S” um	“SIMATIC 300(1)” wird in “ET 200S” umbenannt.
5	Navigieren Sie im SIMATIC-Manager zur ET 200S-Station. Doppelklicken Sie auf das Symbol **Hardware** im rechten Teil des Fensters, um den Editor zum Editieren der Hardware-Konfiguration zu öffnen.	
6	Wenn im rechten Teil des Fensters kein Katalog angezeigt wird, aktivieren Sie diesen, indem Sie im Menü **Ansicht** den Befehl **Katalog** wählen. Navigieren Sie im Katalog über **PROFIBUS-DP** zu **ET 200S.** Fügen Sie dasjenige **IM 151-7 CPU** mit Drag & Drop in das Fenster links oben ein, dessen Bestellnummer mit der Bestellnummer auf Ihrem IM 151-7 CPU übereinstimmt. Standardmäßig wird IM 151-7 CPU als stand-alone-CPU (MPI/ nicht vernetzt) eingebunden. **Hinweis:** Die Bestellnummer im Katalog können Sie erkennen, wenn Sie im Katalog ein **IM 151-7 CPU** mit Mausklick selektieren. Die Bestellnummer dieses IM 151-7 CPU erscheint dann im Feld unter dem Katalog.	

Abfolge	Tätigkeit	Ergebnis
7	Navigieren Sie über das passende **IM 151-7 CPU** zu **PM**. Fügen Sie dasjenige **PM** mit Drag & Drop auf Steckplatz 4 ein, dessen Bestellnummer mit der Bestellnummer auf Ihrem PM übereinstimmt.	
8	Verfahren Sie sinngemäß zu Punkt 8 mit dem DI (auf Steckplatz 5) und dem DO (auf Steckplatz 6)	
9	Wählen Sie im Menü **Station** den Befehl **Speichern und übersetzen.**	Die Hardwarekonfiguration wird übersetzt und gespeichert
10	Laden Sie über MPI die Konfiguration in das IM 151-7 CPU und schließen den Hardware-Editor.	Die Konfiguration ist geladen und im SIMATIC-Manager erscheint im rechten Teil des Fensters ein Symbol für das IM 151-7 CPU.

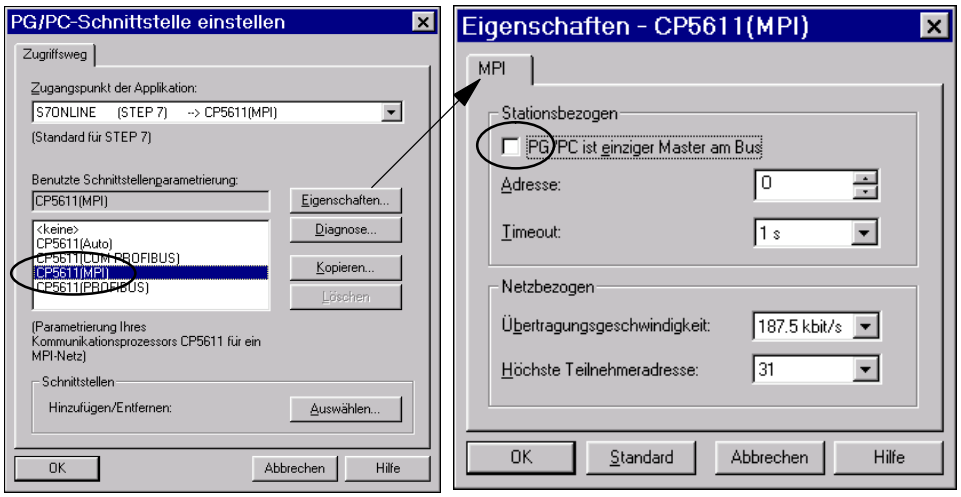
2.5 5. Schritt: Programmieren des IM 151-7 CPU

Abfolge	Tätigkeit	Ergebnis
1	Navigieren Sie im SIMATIC-Manager über **IM151-7 CPU** und **S7-Programm** in den Container **Bausteine**.	
2	Doppelklicken Sie auf das **OB 1**-Symbol im rechten Teil des Fensters.	Es öffnet sich der KOP/FUP/AWL-Editor zum Editieren des Bausteines OB 1.
3	Wählen Sie im KOP/FUP/AWL-Editor im Menü **Ansicht** den Befehl **KOP**, um auf die Programmiersprache KOP umzuschalten.	Im Netzwerk 1 wird ein Strompfad angezeigt.
4	Klicken Sie genau auf die waagrechte Linie des Strompfades.	Die Linie wird hervorgehoben dargestellt.
5	Klicken Sie in der Symbolleiste zweimal auf das Symbol –||– (Schließer) und danach einmal auf das Symbol –( ) (Spule).	Die Symbole werden in den Strompfad eingefügt.
6	Klicken Sie auf die rot dargestellten Fragezeichen des linken Schließers im Strompfad.	Der Schließer wird hervorgehoben dargestellt und an der Stelle der Fragezeichen erscheint ein Text-Eingabefeld mit Cursor.
7	Geben Sie *E1.0* ein und drücken Sie *Return*.	Der linke Schließer erhält die Bezeichnung *E1.0*.
8	Geben Sie *E1.1* ein und betätigen Sie *Return*. Geben Sie *A2.0* ein und betätigen Sie *Return*.	Der rechte Schließer erhält die Bezeichnung *E1.1*. Die Spule erhält die Bezeichnung *A2.0.*
9	Schließen Sie den Editor und beantworten Sie die Frage nach dem Speichern mit **Ja**.	Der Editor wird geschlossen und der OB 1 wird gespeichert.

2.6 6. Schritt: Probelauf

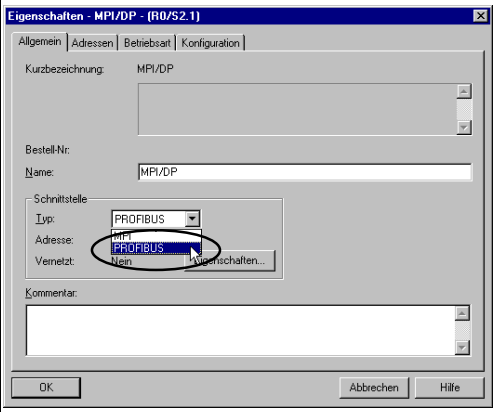
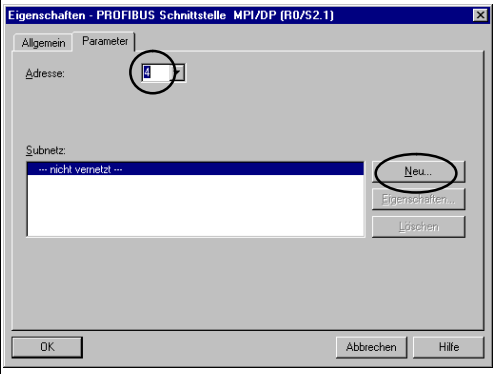
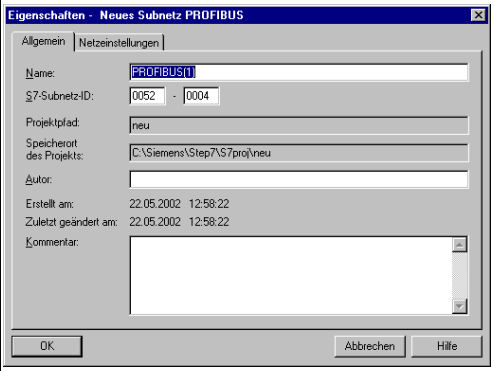
Abfolge	Tätigkeit	Ergebnis
1	Klicken Sie im SIMATIC-Manager auf **Bausteine** im linken Teil des Fensters.	**Bausteine** wird hervorgehoben dargestellt.
2	Klicken Sie mit der rechten Maustaste in den rechten Teil des Fensters und fügen Sie einen leeren Organisationsbaustein mit dem Namen *OB 82* in den Baustein-Container ein. Dieser Baustein stellt sicher, dass die S7-300-CPU auch dann startet, wenn das IM 151-7 CPU noch einen Diagnose-Fehler signalisiert. Generieren Sie auf die gleiche Weise den OB 86. **Hinweis:** Der OB 86 ist erst im DP-Slave-Betrieb von Bedeutung.	Neben dem Baustein OB 1 erscheinen die Bausteine OB 82 und OB 86.
3	Selektieren Sie erneut den Baustein-Container im linken Teil des Fensters. Wählen Sie im Menü **Zielsystem** den Befehl **Laden**, um das Programm und die Hardwarekonfiguration in das IM 151-7 CPU zu übertragen. Quittieren Sie alle erscheinenden Fenster mit **Ja**.	Programm und Konfiguration werden vom PG in das IM 151-7 CPU geladen.
4	Stellen Sie den Betriebsschalter des IM 151-7 CPU auf *RUN*.	Die *STOP*-LED erlischt. Die *RUN*-LED beginnt zu blinken und geht in Dauerlicht über.
5	Betätigen Sie beide Taster abwechselnd.	Die LEDs der Eingänge E1.0 und E1.2 leuchten abwechselnd. Die LED des Ausganges 2.0 leuchtet nicht.
6	Betätigen Sie beide Taster gleichzeitig.	Die LEDs der Eingänge E1.0 und E1.2 (*1*- und *5*-LED der DI) leuchten gleichzeitig. Da im Programm die beiden Taster über eine UND-Funktion (= Reihenschaltung) verknüpft und dem Ausgang A2.0 zugewiesen wurden, leuchtet die LED des Ausganges 2.0 (*1*-LED des DO). Ein angeschlossenes Stellglied oder eine Anzeige wären damit eingeschaltet.
7	Schalten Sie den Betriebsartenschalter des IM 151-7 CPU auf *STOP* und schalten Sie die PS des IM 151-7 CPU aus.	Alle LEDs erlöschen.

2.7 7. Schritt: Umrüsten des IM 151-7 CPU als DP-Slave und in Betrieb nehmen der S7-300

Abfolge	Beschreibung
1	Entfernen Sie den Stecker des PG-Kabels vom IM 151-7 CPU.
2a	Starten Sie, wie unter Schritt 3 Punkt 4 beschrieben, das Programm PG/PC-Schnittstelle einstellen. Ändern Sie die Konfiguration der PG/PC-Schnittstelle wie folgt:
2b	
3	Bestätigen Sie die Einstellungen mit OK und schließen Sie das Programm PG/PC-Schnittstelle einstellen.

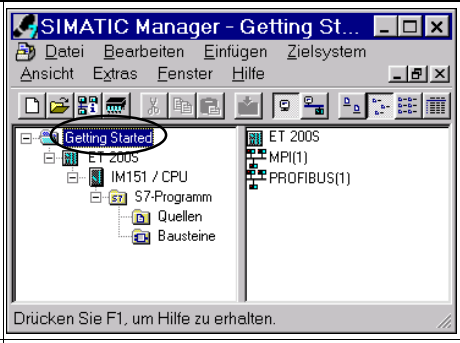
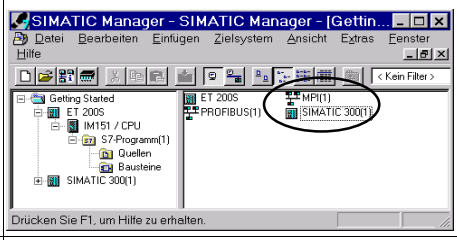
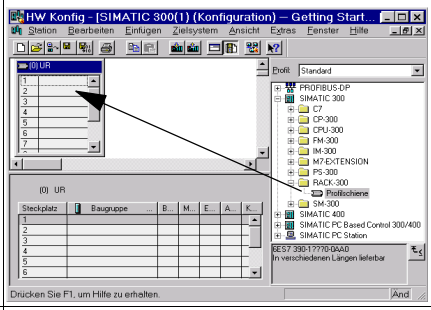
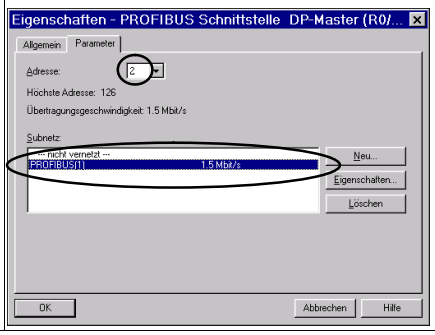
2.8 8. Schritt: Konfigurieren des IM 151-7 CPU als DP-Slave und der S7-300 als DP-Master

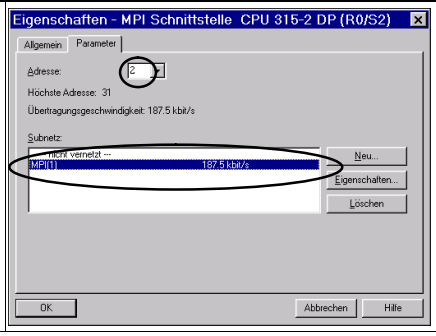
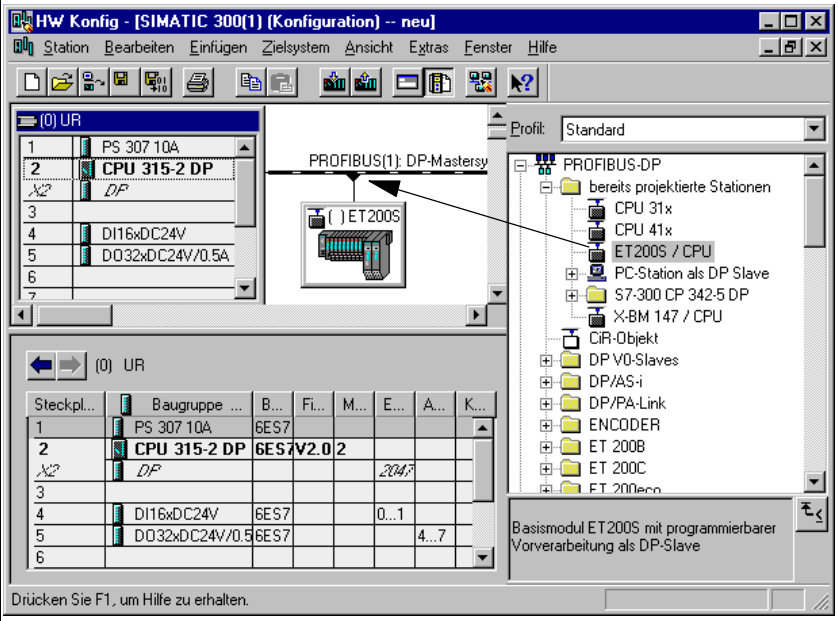
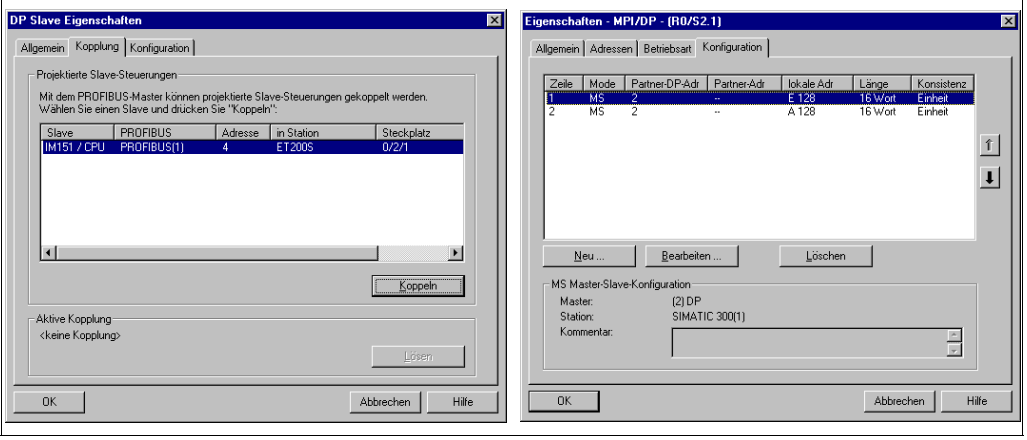
Ändern Sie die Konfiguration des IM 151-7 CPU wie folgt:

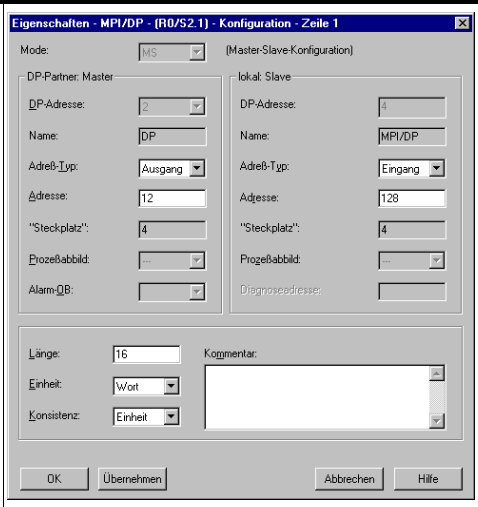
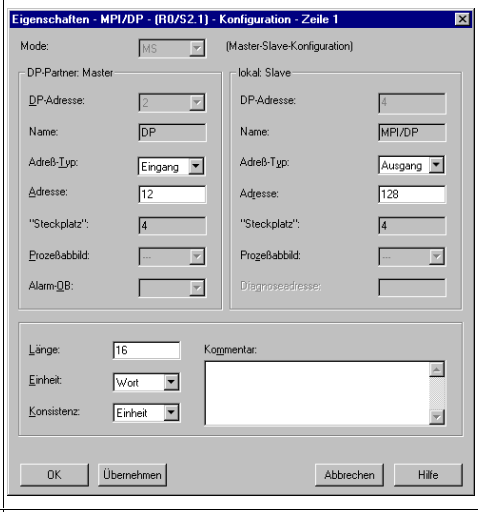
Abfolge	Tätigkeit	Ergebnis
1	Starten Sie wie im Schritt 4 beschrieben das Hardware-Kofigurationsprogramm für das IM 151-7 CPU.	Es öffnet sich der Editor zum Editieren der Hardware-Konfiguration.
2	Wählen Sie im Menü **Eigenschaften – MPI/DP** den Schnittstellentyp **PROFIBUS** aus.	
3	Dann öffnet sich das Fenster **Eigenschaften – PROFIBUS Schnittstelle MPI/DP**. • Stellen Sie die Slave-Adresse **4** ein. • Nach Drücken auf die Schaltfläche **Neu** öffnet sich das Fenster **Neues Subnetz PROFIBUS**.	
4	Im Fenster **Eigenschaften – Neues Subnetz PROFIBUS** überprüfen Sie die Einstellungen und bestätigen mit **OK**.	

Abfolge	Tätigkeit	Ergebnis
5	Wählen Sie im Menü **Station** den Befehl **Speichern und übersetzen.**	Die Hardwarekonfiguration wird übersetzt und gespeichert.
6	Laden Sie über MPI die Konfiguration in das IM 151-7 CPU und schließen den Hardware-Editor.	Die ET 200S hat jetzt die DP-Adresse 4; der Editor wird geschlossen.

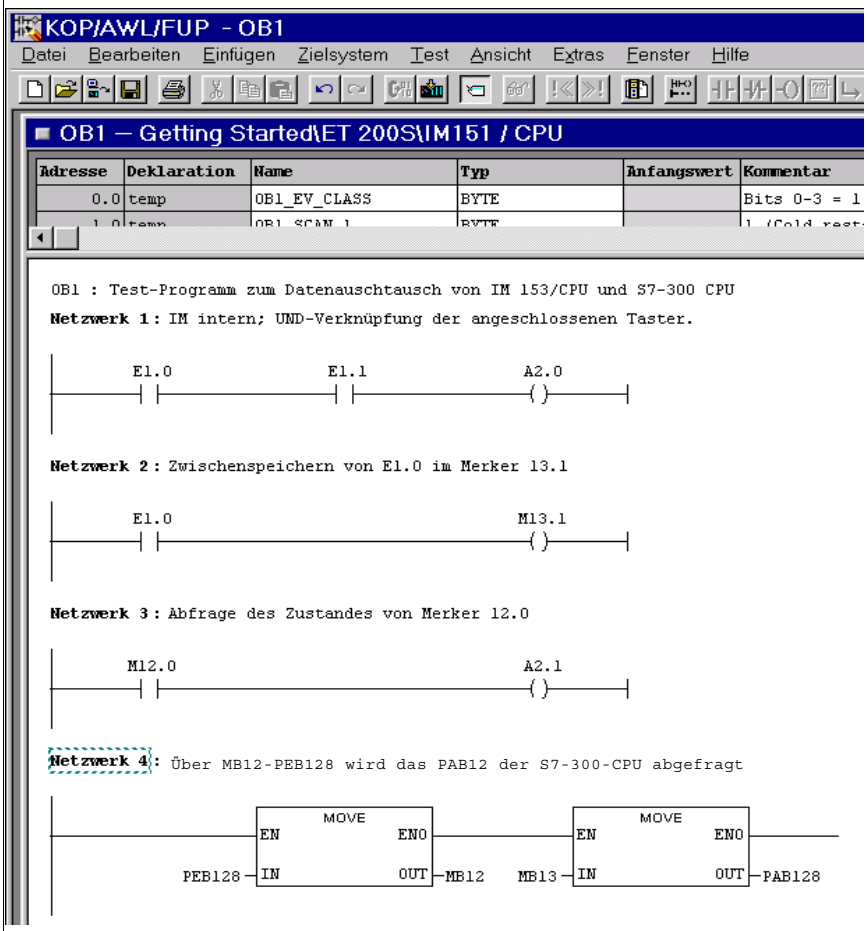
Konfigurieren Sie die S7-300-CPU wie folgt:

Abfolge	Tätigkeit	Ergebnis
1	Selektieren Sie im SIMATIC-Manager im linken Teil des Fensters das Projekt **Getting Started**.	
2	Fügen Sie eine neue S7-300-Station in das Projekt ein, wie im Schritt 4 Punkt 3 beschrieben.	
3	Klicken Sie im SIMATIC-Manager auf die S7-300(1)-Station im linken Teil des Fensters.	Im rechten Teil des Fensters erscheint das Symbol **Hardware.**
4	Doppelklicken Sie auf das Symbol **Hardware** im rechten Teil des Fensters.	Es öffnet sich der Editor zum Editieren der Hardware.
5	Wenn im rechten Teil des Fensters kein Katalog angezeigt wird, aktivieren Sie diesen, indem Sie im Menü **Ansicht** den Befehl **Katalog** wählen. Navigieren Sie im Katalog über **SIMATIC 300** zu **Rack 300.** Fügen Sie eine Profilschiene mit Drag & Drop in das Fenster links oben ein.	
6	Fügen Sie wie in Schritt 4 beschrieben auf Steckplatz 1 die PS ein, deren Bestellnummer mit der Bestellnummer Ihrer PS übereinstimmt. Verfahren Sie in gleicher Weise mit der S7-300-CPU (Steckplatz 2), der S7-300-DI (Steckplatz 4) und der S7-300-DO (Steckplatz 5). **Hinweis:** • Beim Einfügen der S7-300-CPU erscheint ein Fenster. Dort wählen Sie “PROFIBUS-Netz” aus und stellen die Adresse *2* ein. Bestätigen Sie mit **OK**.	Beispiel für die Konfiguration (kann von Ihrer Konfiguration abweichen): 

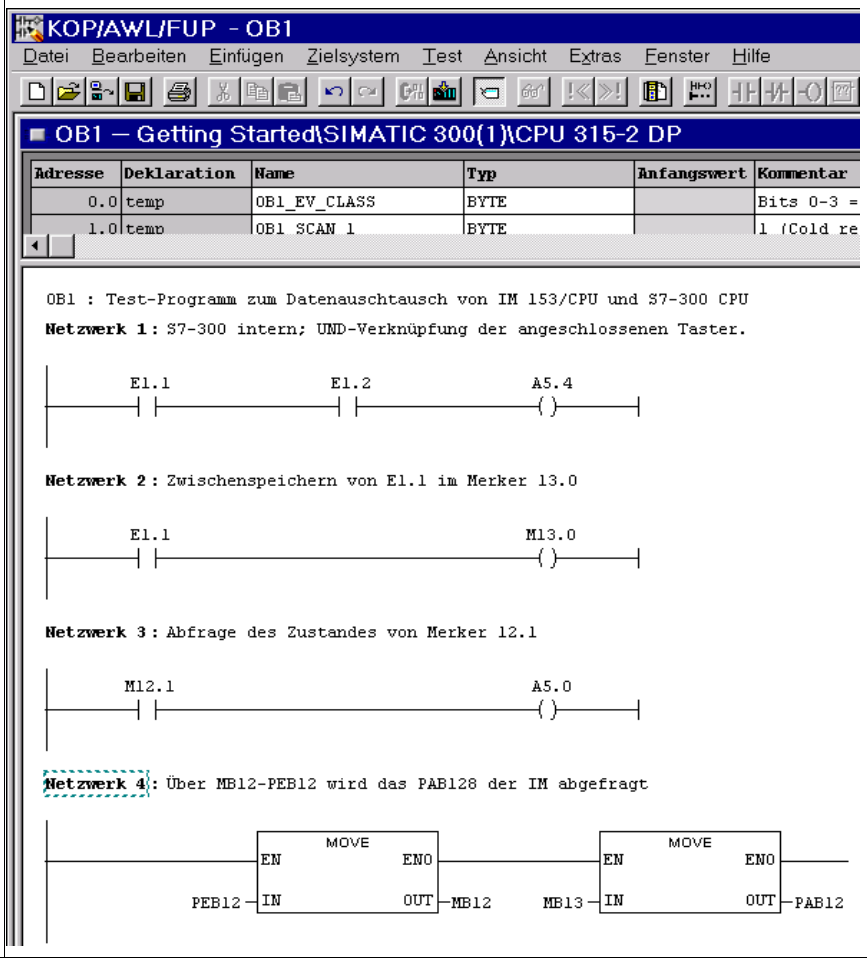
Abfolge	Tätigkeit	Ergebnis
7	• Doppelklicken Sie im linken unteren Teil des Fensters des Hardware-Konfigurationsprogrammes auf **CPU 315-2 DP** (Zeile **2**). • Klicken Sie im nun erscheinenden Fenster auf der Registerkarte **Allgemein** auf die Schaltfläche **Eigenschaften**. • Überprüfen Sie im nun sich öffnenden Fenster “MPI Netz” ob Adresse *2* eingestellt ist. Wenn nein, einstellen. Bestätigen Sie mit **OK**.	
8	Navigieren Sie im Katalog über **PROFIBUS-DP** zu **bereits projektierten Stationen**. Hängen Sie per Drag & Drop die Station **ET 200S/CPU** an das **PROFIBUS-Mastersystem**.	
9	In dem nun erscheinenden Fenster klicken Sie auf die Schaltfläche **Koppeln**. Es erscheint das Fenster **Eigenschaften – MPI/DP**.	

Abfolge	Tätigkeit	Ergebnis
10	Im Fenster von Punkt 9 drücken Sie auf die Schaltfläche **Bearbeiten** und vervollständigen für Zeile 1 die Maske wie im Bild angegeben. Bestätigen Sie danach mit **OK**. Klicken Sie danach im Fenster **Eigenschaften – MPI/DP** auf die zweite Zeile und vervollständigen für Zeile 2 die Maske wie im Bild angegeben. Bestätigen Sie danach mit **OK**.	 
11	Wählen Sie im Menü **Station** den Befehl **Speichern und übersetzen.**	Die Hardwarekonfiguration wird übersetzt und gespeichert.
12	Verbinden Sie das PG über ein PG-Kabel mit der MPI-Schnittstelle der S7-300-CPU. Laden Sie die Konfiguration in die CPU. Schließen Sie den Hardware-Editor.	Die Hardwarekonfiguration wird geladen. Der Editor wird geschlossen.

2.9 9. Schritt: Programmieren des IM 151-7 CPU und der S7-300-CPU

Abfolge	Tätigkeit	Ergebnis
1	Navigieren Sie im SIMATIC-Manager zum Baustein-Container der ET 200S. Doppelklicken Sie auf das **OB1**-Symbol im rechten Teil des Fensters.	Es öffnet sich der KOP/FUP/AWL-Editor zum Editieren des Bausteines OB 1.
2	Ergänzen Sie den OB 1 des IM 151-7 CPU wie folgt: 	
3	Navigieren Sie im SIMATIC-Manager zum Baustein-Container der S7-300. Doppelklicken Sie auf das **OB1**-Symbol im rechten Teil des Fensters.	Es öffnet sich der KOP/FUP/AWL-Editor zum Editieren des Bausteines OB 1.

Abfolge	Tätigkeit	Ergebnis
4	Ergänzen Sie den OB 1 der S7-300 CPU wie folgt:	



Funktionsweise: Der Zustand des an E1.1 der S7-300 angeschlossenen Tasters wird abgefragt und im Merker M13.0 zwischengespeichert. Das gesamte Merkerbyte MB13 wird zum Peripherie-Ausgangsbyte PAB12 transferiert. In der Hardware-Konfiguration haben Sie in Schritt 8 – Konfigurieren der S7-300 Punkt 10 – festgelegt, dass der Bereich von PAW12 bis PAW44 der S7-300 CPU dem Bereich von PEW128 bis PEW160 des IM 151-7 CPU zugeordnet wird.

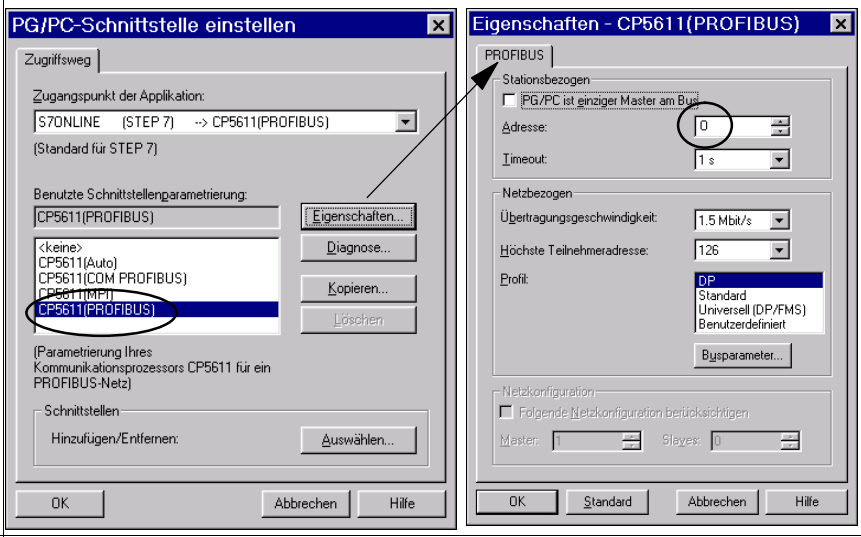
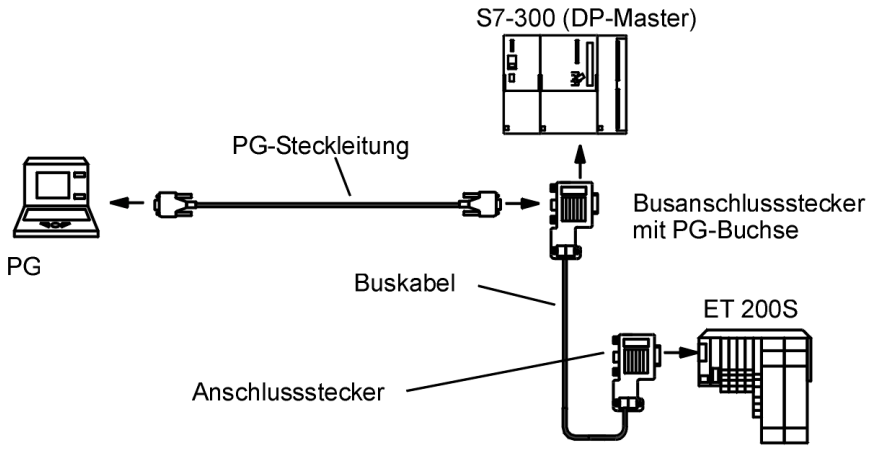
Im Programm des IM wird das PEB128 in das Merkerbyte MB12 tranferiert. Der Merker M12.0 steuert schließlich den Ausgang A2.1 an.

Es ergeben sich damit folgende Kommunikationswege:

S7-300	IM 151-7 CPU
E1.1 → M13.0 MB13 → PAB12 PAW12 →	→ PEW128 PEB128 → MB12 M12.0 → A2.1
	M13.1 ← E1.0 PAB128 ← MB13 PEW12 ← PAW128
MB12 ← PEB12 A5.0 ← M12.1	

2.10 10. Schritt: In Betrieb nehmen und Probelauf von IM 151-7 CPU und S7-300

Abfolge	Tätigkeit	Ergebnis
1	Navigieren Sie im SIMATIC-Manager zum Baustein-Container der S7-300 und fügen Sie einen leeren Organisationsbaustein mit dem Namen *OB 86* in den Baustein-Container ein. Dieser Baustein stellt sicher, dass die S7-300-CPU bei Stationsausfall/-wiederkehr des IM 151-7 CPU nicht in STOP geht. Generieren Sie auf die gleiche Weise den OB 82.	
2	Stellen Sie sicher, dass sich die Betriebsartenschalter von S7 und IM in der *STOP*-Position befinden Schalten Sie die PS der S7-300 und die PS der ET 200S ein.	IM 151-7 CPU und S7-300-CPU fordern Urlöschen an.
3	Urlöschen Sie das IM 151-7 CPU und die S7-300-CPU wie folgt: • Drücken/Drehen Sie den Betriebsartenschalter nach *MRES*. Halten Sie den Betriebsartenschalter in dieser Stellung, bis die *STOP*-LED zum 2. Mal aufleuchtet und dauerhaft leuchtet (entspricht 3 Sekunden). Lassen Sie dann den Betriebsartenschalter los. • Innerhalb von 3 Sekunden müssen Sie den Betriebsartenschalter wieder nach *MRES* drücken/drehen. Die *STOP*-LED beginnt schnell zu blinken und die CPU führt ein Urlöschen durch. Jetzt können Sie den Betriebsartenschalter loslassen. Wenn die *STOP*-LED wieder in Dauerlicht übergeht, hat die CPU das Urlöschen beendet.	Beide CPUs werden urgelöscht.
4	Wählen Sie im SIMATIC-Manager im Menü **Zielsystem** den Befehl **Laden**, um das Programm und die Hardwarekonfiguration in die S7-300-CPU zu übertragen. Quittieren Sie alle erscheinenden Fenster mit **Ja**.	Programm und Konfiguration werden vom PG in die CPU geladen.

Abfolge	Tätigkeit	Ergebnis
5	Starten Sie wie unter Schritt 3 Punkt 4 beschrieben das Programm **PG/PC-Schnittstelle einstellen**. Ändern Sie die Konfiguration der PG/PC-Schnittstelle wie folgt: 	
6	Bestätigen Sie die Einstellungen mit OK und schließen Sie das Programm PG/PC-Schnittstelle einstellen.	
7	Öffnen Sie die Frontklappe der S7-300-CPU. Verbinden Sie das IM 151-7 CPU mit der DP-Schnittstelle der S7-300-CPU über ein PROFIBUS-DP-Kabel. Achten Sie darauf, dass an beiden Steckern der Abschlusswiderstand eingeschaltet ist. Entfernen Sie den Stecker des PG-Kabels von der MPI-Schnittstelle der S7-300-CPU und stecken Sie ihn auf den Busanschluss-Stecker des PROFIBUS-DP-Kabels an der S7-300-CPU. Schrauben Sie den Stecker fest. Schließen Sie die Frontklappe der S7-300-CPU soweit möglich. 	

Abfolge	Tätigkeit	Ergebnis
8	Navigieren Sie im SIMATIC-Manager zum Baustein-Container der ET 200S. Selektieren Sie den Baustein-Container im linken Teil des Fensters. Wählen Sie im SIMATIC-Manager im Menü **Zielsystem** den Befehl **Laden**, um das Programm und die Hardwarekonfiguration in das IM 151-7 CPU zu übertragen. Quittieren Sie alle erscheinenden Fenster mit **Ja**.	Programm und Konfiguration werden vom PG in das IM 151-7 CPU geladen.
9	Stellen Sie den Betriebsartenschalter des IM 151-7 CPU auf *RUN*.	Die *STOP*-LED des IM erlischt. Die *RUN*-LED beginnt zu blinken und geht in Dauerlicht über. Die *SF*-LED leuchtet.
10	Stellen Sie den Betriebsartenschalter der S7-300-CPU auf *RUN*.	Die *STOP*-LED der S7 erlischt. Die *RUN*-LED beginnt zu blinken und geht in Dauerlicht über. Die *SF*-LED der IM erlischt.
11	Betätigen Sie beide Taster der S7-300 abwechselnd.	Die LEDs der S7-300-Eingänge E1.1 und E1.2 leuchten abwechselnd. Die LED des Ausgangs 5.4 leuchtet nicht.
12	Betätigen Sie beide Taster der S7-300 gleichzeitig.	Die LEDs der Eingänge E1.1 und E1.2 leuchten gleichzeitig. Da im Programm die beiden Taster über eine UND-Funktion (= Reihenschaltung) verknüpft und dem Ausgang A5.4 zugewiesen wurden, leuchtet die LED des Ausgangs 5.4.
13	Betätigen Sie den Schalter, der an E1.0 der ET 200S angeschlossen ist.	Die LEDs des IM-Eingangs E1.0 und des S7-300-Ausgangs A5.0 leuchten.
14	Betätigen Sie den Schalter, der an E1.1 der S7-300 angeschlossen ist.	Die LEDs des S7-300-Eingangs E1.1 und des IM-Ausgangs A2.1 leuchten.

Diagnose und Beheben von Störungen

Durch falsche Bedienung, inkorrektes Verdrahten oder falsche Hardwarekonfiguration können Fehler auftreten, welche die CPU nach dem Urlöschen mit der Sammelfehler-LED *SF* anzeigt.

Wie Sie solche Fehler und Meldungen diagnostizieren können, ist beschrieben in den Handbüchern

- *Installationshandbuch S7-300;* Kapitel 10.4
- *Programmieren mit STEP 7 V5.1*; Kapitel 21
- *Interfacemodul IM 151-7 CPU*; Kapitel 5

Weiterführende Handbücher

Als ergänzendes Getting Started empfehlen wir: *Getting Started Erste Schritte und Übungen mit STEP 7 V5.1*.

Alle Handbücher können Sie kostenlos von der Homepage der Firma Siemens (Customer Support Automatisierungstechnik) herunterladen.

Adressierung

3

Prinzip des Datenaustauschs zwischen DP-Master und IM 151-7 CPU

In diesem Kapitel finden Sie Informationen über die Adressierung von Peripheriemodulen und den Datenaustausch zwischen DP-Master und IM 151-7 CPU.

Bei der Adressierung der Peripheriemodule haben Sie folgende Alternativen:

- Steckplatzorientierte Adressvergabe:
  Die steckplatzorientierte Adressvergabe entspricht der Defaultadressierung, d. h. STEP 7 ordnet jeder Steckplatznummer eine festgelegte Baugruppen-Basisadresse zu.
- Freie Adressvergabe:
  Sie können jeder Baugruppe eine beliebige Adresse innerhalb des erlaubten Adressbereiches des IM 151-7 CPU zuordnen.

Zur Adressierung des IM 151-7 CPU am PROFIBUS-DP siehe Kapitel 4.3.

Kapitelübersicht

Kapitel	Thema	Seite
3.1	Steckplatzorientierte Adressierung der Peripheriemodule	3-2
3.2	Freie Adressierung der Peripheriemodule	3-4
3.3	Datenaustausch mit dem DP-Master	3-5
3.4	Zugriff auf den Übergabespeicher im IM 151-7 CPU	3-7

3.1 Steckplatzorientierte Adressierung der Peripheriemodule

Steckplatzorientierte Adressvergabe

Bei der steckplatzorientierten Adressierung (Defaultadressierung) ist jeder Steckplatznummer eines Moduls ein Adressbereich im IM 151-7 CPU zugeordnet.

Abhängig vom Typ des Peripheriemoduls gibt es Digital- und Analogadressen (siehe Tabelle 3-1). Die Adresszuordnung ist nicht fest und kann geändert werden, allerdings gibt es einen defaultmäßig eingestellten Adressraum.

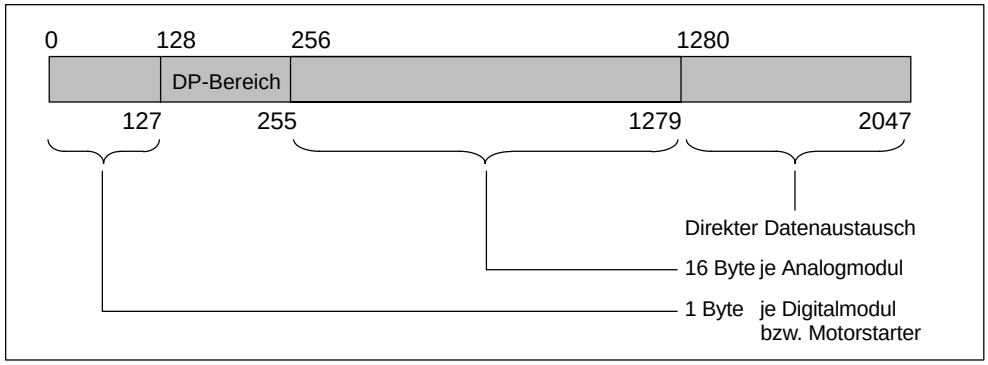


Bild 3-1 Aufbau des defaultmäßig eingestellten Adressraums

Steckplatzzuordnung

Das folgende Bild zeigt einen Aufbau eines ET 200S mit digitalen Elektronikmodulen, analogen Elektronikmodulen, Technologiemodulen und der Steckplatzzuordnung.

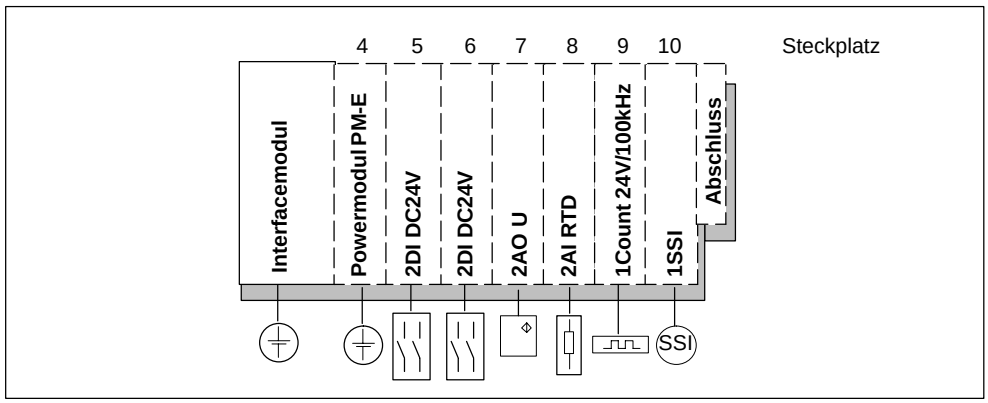


Bild 3-2 Steckplätze bei ET 200S

Adresszuordnung

Für jedes der maximal 63 Peripheriemodule sind, in Abhängigkeit vom Steckplatz, in den Adressbereichen des IM 151-7 CPU 1 Byte für Digitalperipherie und 16 Byte für Analogperipherie reserviert.

In der folgenden Tabelle finden Sie die Default-Adresszuordnung für Analog- und Digitalmodule pro Steckplatz. Die Adressbereiche der Peripheriemodule sind nur für ein IM 151-7 CPU innerhalb des ET 200S “sichtbar” und nicht für den zugehörigen DP-Master. Der DP-Master hat keinen direkten Zugriff auf die Peripheriemodule.

Tabelle 3-1 Adressen der Peripheriemodule von ET 200S

reservierter Adressbereich	**Steckplatznummer**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**...**	**66**
Digitalmodule, Motorstarter	IM 151-7 CPU			-	1	2	3	4	...	62
Analogmodule, Technologiemodule				-	272 bis 287	288 bis 303	304 bis 319	320 bis 335	...	1248 bis 1263
Powermodule				256	272	288	304	320		1248

Die nicht belegten Adressen im Bereich 64 bis 127 liegen bei Defaultadressierung im Prozessabbild und können im Anwenderprogramm frei verwendet werden. Sind in einem Byte bereits 2 Bit durch eine Digitalbaugruppe belegt, so sind die restlichen 6 Bit nicht verwendbar (z. B. im Bild 3-3 die Bits 1.4 bis 1.7).

Die nicht durch Module belegten Bytes der Adressbereiche können Sie in Ihrem Anwenderprogramm frei verwenden. Bei der Konfiguration im Bild 3-3 sind beispielsweise die Bytes 2 und 3 frei verwendbar.

Beispiel für die steckplatzorientierte Adresszuordnung zu Peripheriemodulen

Im folgenden Bild sehen Sie einen Beispiel-Aufbau für ein ET 200S, an dem die Adressvergabe für die Peripheriemodule exemplarisch gezeigt wird. Die Adressen für die Peripheriemodule sind bei Defaultadressierung fest vorgegeben.

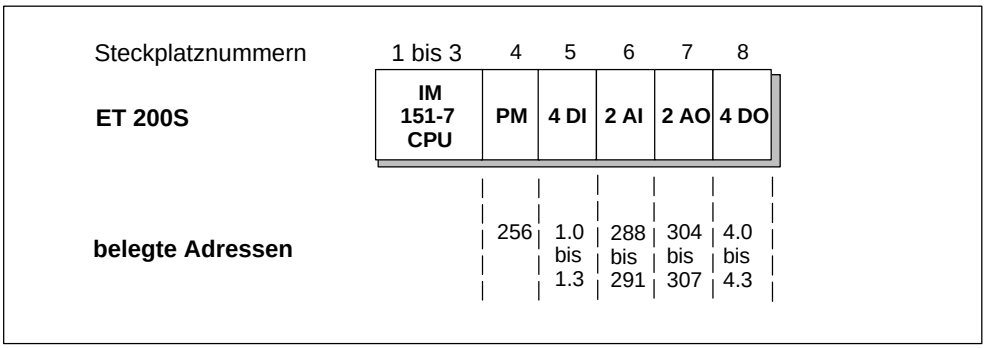


Bild 3-3 Beispiel für die Adresszuordnung zu Peripheriemodulen

3.2 Freie Adressierung der Peripheriemodule

Freie Adressvergabe

Freie Adressvergabe heißt, sie können

- Eingangsadressen von Baugruppen und
- Ausgangsadressen von Baugruppen

im Bereich von 0 bis 2047 bytegranular und von einander unabhängig frei wählen. Die Adressen 0 bis 127 liegen im Prozessabbild. Die Adresszuordnung nehmen Sie in *STEP 7* vor. Sie legen dabei die Basisadresse der Baugruppe fest, auf der dann alle weiteren Adressen der Baugruppe basieren.

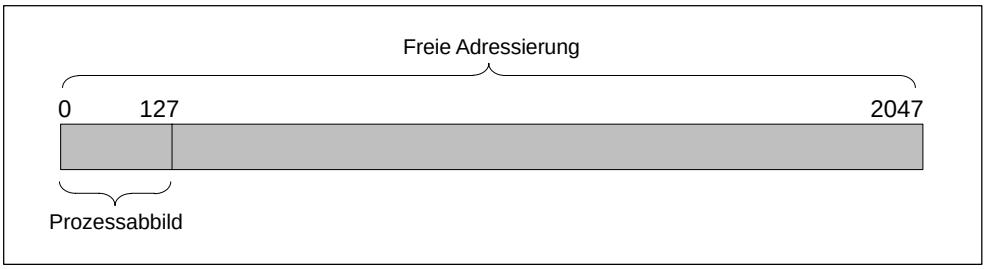


Bild 3-4 Aufbau des Adressraums bei freier Adressierung

Hinweis

In der freien Adressvergabe ist keine bitgranulare Adressierung möglich, daher wird eine Komprimierung von Digitalkanälen nicht unterstützt. Ein “Packen” von Adressen ist nicht möglich.

Vorteil

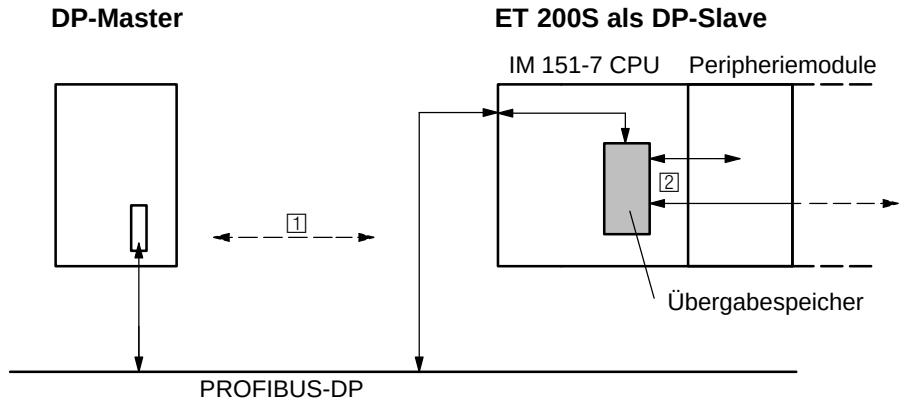
Vorteile der freien Adressvergabe:

- Sie können die verfügbaren Adressräume optimal nutzen, da keine “Adresslükken” zwischen den Baugruppen bleiben.
- Bei der Erstellung von Standardsoftware können Sie Adressen angeben, die unabhängig von der jeweiligen Konfiguration der ET 200S-Station sind.

3.3 Datenaustausch mit dem DP-Master

Nutzdatentransfer über einen Übergabespeicher

Die Nutzdaten liegen in einem Übergabespeicher im IM 151-7 CPU. Der Nutzdatentransfer zwischen dem IM 151-7 CPU und dem DP-Master erfolgt immer über diesen Übergabespeicher. Der Übergabespeicher besteht aus maximal 32 Adressbereichen.



1 Datenaustausch zwischen DP-Master und ET 200S über Übergabespeicher im IM 151-7 CPU
2 Datenaustausch zwischen IM 151-7 CPU und Peripheriemodulen

Bild 3-5 Prinzip des Datenaustauschs zwischen DP-Master und ET 200S mit IM 151-7 CPU

Adressbereiche für Nutzdatentransfer mit DP-Master

Das ET 200S stellt für PROFIBUS-DP maximal 244 Byte Ein- und 244 Byte Ausgangsdaten zur Verfügung. Die Adressierung dieser Daten im Übergabespeicher des IM 151-7 CPU ist in bis zu 32 Adressbereichen möglich.

Ein Adressbereich umfasst dabei maximal 32 Byte. Insgesamt stehen jeweils maximal 244 Byte für Ein- und Ausgangsdaten zur Verfügung.

Die Adressbereiche beginnen defaultmäßig bei 128. Die Daten werden lückenlos ab Adresse 128 eingetragen.

Datenkonsistenz

Sie definieren die Datenkonsistenz als Byte-, Wort- oder Gesamtkonsistenz pro Adressbereich. Die Konsistenz darf pro Adressbereich maximal 32 Byte/16 Worte betragen.

DP-Diagnoseadresse in *STEP 7*

Bei der Projektierung des ET 200S mit *STEP 7* werden zwei Diagnoseadressen eingestellt. Über diese Diagnoseadressen erhält das ET 200S Auskunft über den Zustand des DP-Masters bzw. über eine Busunterbrechung (siehe Kapitel 6.5). Im DP-Slave-Betrieb liegen die Diagnoseadressen defaultmäßig auf 2045 und 2046.

2045: Adresse für Steckplatz 2 (IM 151-7 CPU)
2046: Diagnoseadresse

Detaillierte Informationen finden Sie in der *Online-Hilfe zu STEP 7* unter *Steckplatzmodell bei I-Slaves*.

Zugriff auf freie Bereiche im Prozessabbild

Wenn Sie auf vorhandene, aber nicht projektierte Prozessabbildbereiche zugreifen, werden keine Prozessabbildfehler erzeugt. Sie können also Ein- und Ausgänge im Prozessabbild, denen keine Peripheriebaugruppen zugeordnet sind, als Merker verwenden.

3.4 Zugriff auf den Übergabespeicher im IM 151-7 CPU

Zugriff im Anwenderprogramm

In der folgenden Tabelle ist dargestellt, wie Sie aus dem Anwenderprogramm heraus auf den Übergabespeicher im IM 151-7 CPU zugreifen können.

Tabelle 3-2 Zugriffe auf die Adressbereiche

Zugriff in Abhängigkeit von der Datenkonsistenz	Es gilt
1, 2 bzw. 4 Byte Datenkonsistenz mit Lade-/Transferbefehlen	Es kann auf alle Bereiche, die mit Konsistenz “Einheit” parametriert wurden, zugegriffen werden. Maximal 64 Byte Eingangsdaten können Sie mit Ladeoperationen und maximal 64 Byte Ausgangsdaten mit Transferoperationen ansprechen (L PEB/PEW/PED; T PAB/PAW/PAD; siehe auch Anhang A). Bei Wortzugriffen ergibt sich eine Datenkonsistenz von 2 Byte und bei Doppelwortzugriffen von 4 Byte. Zugriffe sind auch über das Prozessabbild möglich.
1 bis 32 Byte Datenkonsistenz am PROFIBUS-DP mit SFC 14 und SFC 15	Wenn der Adressbereich konsistenter Daten im Prozessabbild liegt, dann wird dieser Bereich automatisch aktualisiert. Wenn Sie auf Daten im Übergabespeicher zugreifen wollen, dann müssen Sie die Eingangsdaten mit dem SFC 14 “DPRD_DAT” lesen und die Ausgangsdaten mit dem SFC 15 “DPWR_DAT” schreiben. Diese SFCs haben eine Datenkonsistenz von 1 bis 32 Byte. Die mit dem SFC 14 gelesenen Eingangsdaten können Sie nur als Block von 1 bis 32 Byte zum Beispiel in einen Merkerbereich kopieren und dort mit U M x.y ansprechen. Ebenso können Sie nur einen Block von 1 bis 32 Byte mit dem SFC 15 als Ausgangsdaten schreiben (siehe auch Referenzhandbuch *System- und Standardfunktionen*). Beim Zugriff auf Bereiche mit Konsistenz “Gesamte Länge” muss die Länge beim SFC mit der Länge des parametrierten Bereichs übereinstimmen. Außerdem sind auch Direktzugriffe auf die konsistenten Bereiche möglich (z. B. L PEW oder T PAW).

Regeln für die Adressvergabe

Folgende Regeln müssen Sie bei der Adressvergabe für ET 200S mit IM 151-7 CPU beachten:

- Zuordnung der Adressbereiche:
  - Eingangsdaten des ET 200S sind **immer** Ausgangsdaten des DP-Masters
  - Ausgangsdaten des ET 200S sind **immer** Eingangsdaten des DP-Masters
- Im Anwenderprogramm greifen Sie mit Lade-/Transferoperationen bzw. mit den SFCs 14 und 15 auf die Daten zu.
- Die Länge, Einheit und die Konsistenz der zusammengehörenden Adressbereiche für DP-Master und DP-Slave muss gleich sein.
- Adressen für den Master und den Slave können im logisch gleichen Übergabespeicher unterschiedlich sein (voneinander unabhängige logische Peripherieadressräume in der Master- und der Slave-CPU)

Bei der Projektierung des IM 151-7 CPU durch *STEP 7* zum Betrieb in der S5 oder in Fremdsystemen werden natürlich nur die logischen Adressen innerhalb der Slave-CPU vergeben, die Zuordnung im Mastersystem erfolgt mit dem spezifischen Projektierungswerkzeug des Mastersystems.

Adressieroberfläche in *STEP 7*

Die folgende Tabelle zeigt das Prinzip der Adressvergabe. Die Tabelle finden Sie in der *STEP 7*-Oberfläche wieder. Bei *STEP 7* müssen Sie den Mode “MS” für Master-Slave oder “DX” für den Direkten Datenaustausch (siehe Kapitel 4.5) einstellen.

Tabelle 3-3 Adressieroberfläche in *STEP 7 V5.1 (Ausschnitt)*

	Mode	**Master**		**PROFIBUS-DP-Partner**		**Parameter**		
		E/A	Adresse	E/A	Adresse	Länge	Einheit	Konsistenz
1	MS	A	200	E	128	4	Byte	Einheit
2	MS	A	300	E	132	8	Byte	gesamte Länge
3	MS	E	700	A	128	4	Wort	Einheit
4	MS	E	50	A	136	4	Byte	Einheit
	MS: Master Slave	Adressbereiche in der DP-Master-CPU		Adressbereiche im IM 151-7 CPU		Diese Parameter der Adressbereiche müssen für DP-Master und IM 151-7 CPU gleich sein		

Defaulteinstellung für Adressbereiche

Wenn Sie bei der Projektierung des ET 200S keine Adressbereiche für den Datenaustausch mit dem DP-Master parametrieren, dann läuft das ET 200S nach der Inbetriebnahme mit einer Defaulteinstellung am PROFIBUS-DP an.

Die Defaulteinstellung ist:

- 16 Worte Eingangsdaten; Konsistenz Einheit (d. h. Wort)
- 16 Worte Ausgangsdaten; Konsistenz Einheit (d. h. Wort)

Wenn Sie das IM 151-7 CPU stand-alone (MPI/nicht vernetzt) projektieren, erfolgt keine Defaulteinstellung der Adressbereiche, da im Stand-alone-Betrieb kein Übergabespeicher projektiert ist.

Beispielprogramm

Im Folgenden sehen Sie in einem Beispielprogramm den Datenaustausch zwischen DP-Master und DP-Slave.

Sie finden die Adressen aus Tabelle 3-3 wieder.

Der Aufruf der SFCs 14 und 15 erfolgt unter Angabe der logischen Adresse im Hex-Format.

im IM 151-7 CPU

```
Datenvorverarbeitung im DP-Slave:

L       2                           Lade Istwert 2 und
T       MB      6                   transferiere nach Merkerbyte 6.
L       EB      0                   Lade Eingangsbyte 0 und
T       MB      7                   transferiere nach Merkerbyte 7.
```

```
Daten weiterreichen an DP-Master:

L       MW      6                   Lade Merkerwort 6 und
T       PAW     136                 transferiere nach Peripherieausgangswort 136
```

in der DP-Master-CPU

```
empfangene Daten im DP-Master weiterverarbeiten:

L       PEB     50                  Lade Peripherieeingangsbyte 50 und
T       MB      60                  transferiere nach Merkerbyte 60.
L       PEB     51                  Lade Peripherieeingangsbyte 51 und
L       B#16#3                      lade Byte 3;
+       I                           addiere die Werte als Datentyp Integer und
T       MB      61                  transferiere Ergebnis nach Merkerbyte 61.
```

```
Datenvorverarbeitung im DP-Master:

L       10                          Lade Istwert 10 und
+       3                           addiere 3,
T       MB      67                  transferiere Ergebnis nach Merkerbyte 67.
```

```
Daten (Merkerbyte 60 bis 67) senden an DP-Slave:

CALL    SFC     15                  Rufe Systemfunktion 15 auf:
 LADDR:= W#16#12C                   Schreibe in einer Länge von 8 Byte ab Merkerbyte
 RECORD:= P#M60.0 Byte8             60 die Daten in den Ausgangsadressbereich ab
 RET_VAL:= MW 22                    Adresse 300 (12C hex).
```

im IM 151-7 CPU

```
Daten empfangen vom DP-Master (abgelegt in MB 30 bis 37):

CALL    SFC     14                  Rufe Systemfunktion 14:
 LADDR:=W#16#84                     Schreibe die Daten vom Eingangsadressbereich ab
 RET_VAL:=MW 20                     Adresse 132 (84 hex) in einer Länge von 8 Byte
 RECORD:=P#M30.0 Byte8              nach Merkerbyte 30.
```

```
empfangene Daten weiterverarbeiten:

L       MB      30                  Lade Merkerbyte 30 und
L       MB      37                  lade Merkerbyte 37;
+       I                           addiere die Werte als Datentyp Integer und
T       MW      100                 transferiere Ergebnis nach Merkerbyte 100.
```

Nutzdatentransfer im Betriebszustand STOP

Je nachdem, ob der DP-Master oder der DP-Slave (IM 151-7 CPU) in STOP gehen, werden die Nutzdaten im Übergabespeicher verschieden behandelt.

- Das IM 151-7 CPU geht in STOP: Die Daten im Übergabespeicher (nur Ausgänge aus Slavesicht) des IM 151-7 CPU werden mit “0” überschrieben, d. h. der DP-Master bzw. ein Empfänger im direkten Datenaustausch liest “0”.
- Der DP-Master geht in STOP: Die aktuellen Daten im Übergabespeicher des IM 151-7 CPU (Eingänge im Slave, Ausgänge im Master) bleiben erhalten und können im Anwenderprogramm des IM 151-7 CPU ausgelesen werden.

ET 200S im PROFIBUS-Netz

4

Einleitung

Sie können ET 200S mit IM 151-7 CPU als Teilnehmer in ein PROFIBUS-Netz integrieren. In diesem Kapitel finden Sie einen typischen Netz-Aufbau mit IM 151-7 CPU erläutert. Außerdem erfahren Sie, welche Funktionen über ein PG oder OP auf ET 200S ausführbar sind und welche Möglichkeiten Ihnen für den direkten Datenaustausch geboten werden. Die verfügbaren Kommunikationsdienste finden Sie im Kapitel 7.7.

Kapitelübersicht

Kapitel	Thema	Seite
4.1	ET 200S im PROFIBUS-Netz	4-2
4.2	Netzkomponenten	4-6
4.3	PROFIBUS-Adresse	4-8
4.4	Funktionen über PG/OP	4-9
4.5	Direkter Datenaustausch	4-12

Weitere Informationen

Weitere Informationen zum Aufbau von Netzen finden Sie im Handbuch zum DP-Master.

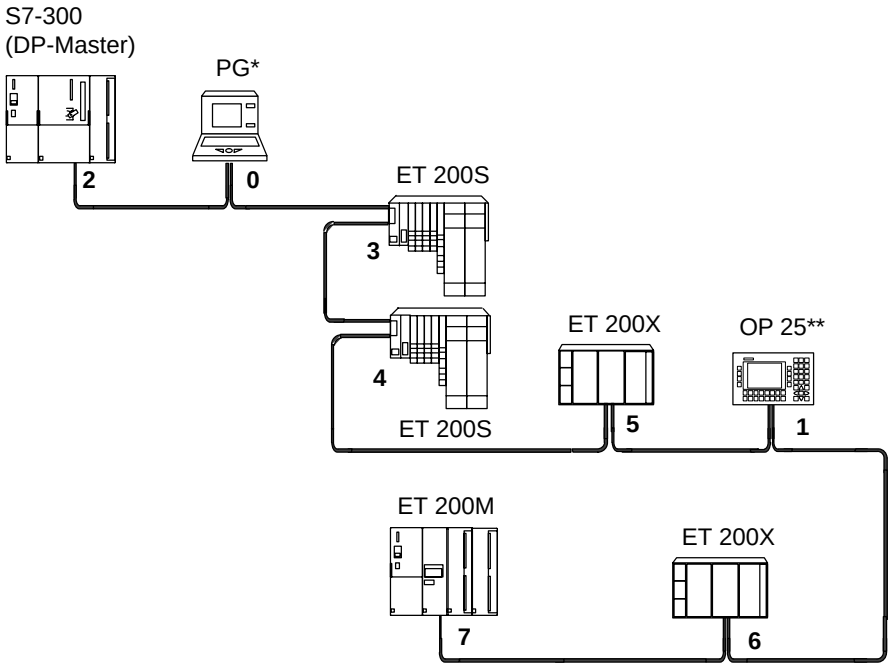
Anschluss der Lichtwellenleiter (LWL) an das IM 151-7 CPU FO

Informationen zum Anschluss der LWL an das IM 151-7 CPU FO finden Sie im Handbuch *Dezentrales Peripheriegerät ET 200S* im Kapitel *Verdrahten und Bestücken*. Der dort für das IM 151 FO beschriebene Sachverhalt gilt analog für das IM 151-7 CPU FO.

4.1 ET 200S im PROFIBUS-Netz

Aufbau eines PROFIBUS-Netzes

Das folgende Bild zeigt Ihnen den prinzipiellen Aufbau eines PROFIBUS-Netzes mit einem DP-Master und mehreren DP-Slaves.



* **von diesem PG aus ist ET 200S projektier- und programmierbar**

** **Bedien- und Beobachtungsfunktionen auf ET 200S ausführbar**

0 ... 7 PROFIBUS-Adressen der Teilnehmer

Bild 4-1 Beispiel für ein PROFIBUS-Netz

Hardware-Voraussetzung im PG/OP für Zugriff auf ET 200S

Damit Sie von einem PG/OP aus auf ein IM 151-7 CPU zugreifen können, muss das PG/OP folgende Voraussetzungen erfüllen:

- mit einer integrierten PROFIBUS-DP-Schnittstelle oder DP-Karte ausgestattet sein oder
- mit einer integrierten MPI-Schnittstelle oder MPI-Karte ausgestattet sein

Zugriff auf ET 200S

Das IM 151-7 CPU ist ein passiver/aktiver Bus-Teilnehmer. Die Programme und Projektierung des IM 151-7 CPU können dabei aus dem SIMATIC Manager mittels “Zielsystem laden” vom PG auf das IM 151-7 CPU übertragen werden. Daneben sind auch alle weiteren Diagnose- und Testfunktionen mit dem PG möglich.

Falls das PG aktuell der einzige aktive Busteilnehmer ist, ist vorher über den Menüpunkt “PG/PC-Schnittstelle einstellen” dieses im SIMATIC Manager entsprechend einzustellen (siehe Kapitel 4.4).

Sie haben weiterhin die Möglichkeit, OPs/OSs (Operator Panels/Operator Stations) für Bedien- und Beobachtungsfunktionen fest im PROFIBUS-Netz zu installieren.

Sie können insgesamt von maximal 12 Geräten aus parallel auf ein ET 200S zugreifen:

- 1 Verbindung ist fest reserviert für das PG.
- 1 Verbindung ist fest reserviert für ein OP oder eine OS.
- 10 Verbindungen sind frei verfügbar für PGs, OPs/OSs, CPUs

Wir empfehlen Ihnen, die PROFIBUS-Adresse für das PG/OP wie für andere Netz-Teilnehmer zu vergeben (siehe Bild 4-1).

Aktive/passive DP-Schnittstelle des IM 151-7 CPU

Die Betriebsart der DP-Schnittstelle am IM 151-7 CPU stellen Sie bei der Projektierung im Fenster **Eigenschaften – MPI/DP** ein:

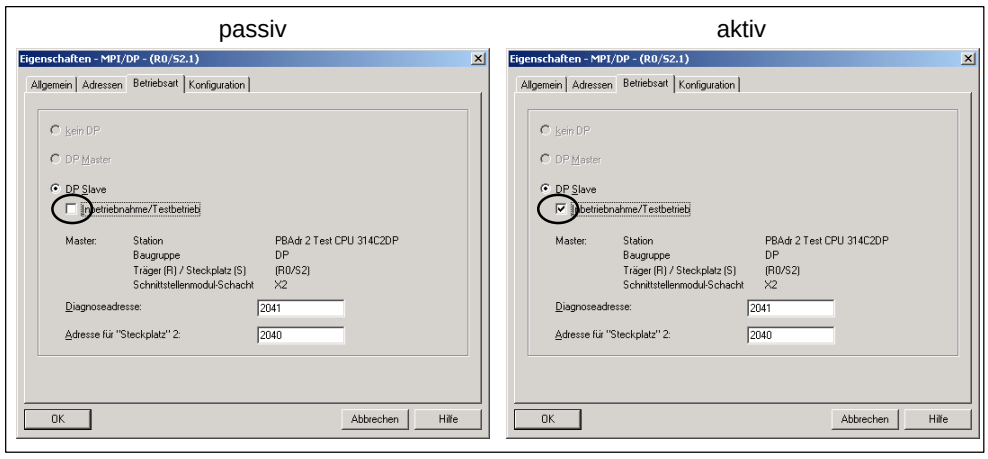


Bild 4-2 Betriebsarteneinstellung der DP-Schnittstelle am IM 151-7 CPU

Abhängig von der vorgenommenen Einstellung der DP-Schnittstelle hat das IM 151-7 CPU folgendes Verhalten:

Tabelle 4-1 Verhalten des IM 151-7 CPU in Abhängigkeit von der Einstellung der DP-Schnittstelle

	DP-Schnittstelle des IM 151-7 CPU	
	passiv	**aktiv**
Baudratensuche	ja	nein
Test- und Inbetriebnahme-funktionen	langsamer	schneller
Busumlaufzeit	schneller	langsamer
Diagnose über BF-LED	siehe Kapitel 6.4	

Nach der Inbetriebnahme empfehlen wir, die DP-Schnittstelle passiv einzustellen.

Maximale Datenübertragungsrate mit einer PG-Steckleitung

Mit der PG-Steckleitung können Sie eine maximale Datenübertragungsrate von 1,5 Mbaud realisieren.

PG-Anschluss bei IM 151-7 CPU FO

Der PG-Anschluß steht nur für einen Teilnehmer (PG oder OP) zur Verfügung und ist **nicht** für eine Vernetzung vorgesehen
Die Busabschlusswiderstände sind im PG-Anschluss der IM 151-7 CPU FO fest integriert.
Stellen Sie deswegen bei Verwendung von Leitungen mit Busanschlusssteckern (bei Baudraten > 1,5 MBaud erforderlich) am Busanschlussstecker an der CPU-Seite den Abschlusswiderstand auf OFF und auf PG-/OP-Seite wie gewohnt auf ON! Bei Verwendung einer PG-Steckleitung (nur bis 1,5 MBaud zulässig) müssen Sie nichts beachten.

Beispiele für den Anschluss PG/OP am ET 200S

- Das PG/OP steckt auf der PROFIBUS-DP-Schnittstelle des DP-Masters, kann aber genau so an jede weitere Station, die an dem DP-Netz hängt, angesteckt werden, ebenso an das ET 200S.

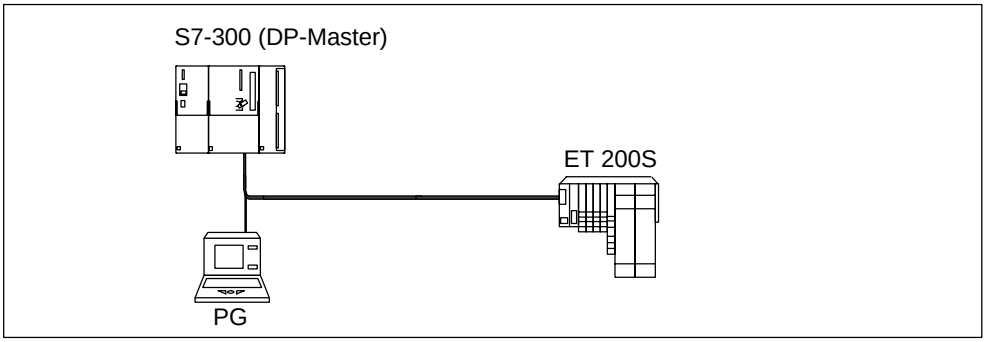


Bild 4-3 PG/OP greift über DP-Schnittstelle im DP-Master auf ET 200S zu

- Das PG ist direkt mit dem ET 200S verbunden (ET 200S fügen Sie erst später in das PROFIBUS-Netz ein).
  Zu beachten: Abhängig von der DP-Schnittstelle (aktiv/passiv) ist eine besondere Einstellung in *STEP 7* Voraussetzung (siehe Kapitel 4.4).

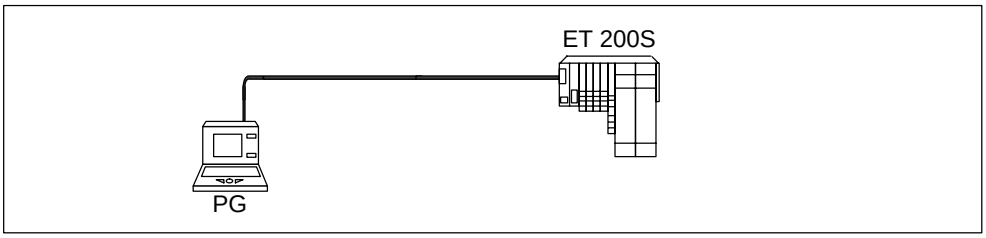


Bild 4-4 PG greift direkt auf ET 200S zu

- Das PG kann auch ein direkter DP-Teilnehmer sein, wobei bei einer Baudrate von 12 Mbaud keine Stichleitung (z. B. PG-Steckleitung) zulässig ist.

4.2 Netzkomponenten

Sie benötigen für den Anschluss des ET 200S an das PROFIBUS-DP-Netz folgende Netzkomponenten:

Tabelle 4-2 Netzkomponenten

Zweck	Netzkomponenten	Bestellnummern
zum Aufbauen des Netzes	Kabel, z. B. 2adrig, geschirmt oder 5adrig, ohne Konfektionierung	6XV1 830-0AH10 (2adrig) 6XV1 830-0BH10 (2adrig mit PE-Mantel) 6XV1 830-3CH10 (2adrig, für Girlandenaufhängung) 6XV1 830-3BH10 (Schleppkabel) 6XV1 830-3AH10 (Erdverlegungskabel) 6ES7 194-1LY00-0AA0-Z (5adrig mit PVC-Mantel) 6ES7 194-1LY10-0AA0-Z (5adrig; ölbeständig, schleppfähig, bedingt schweißfest; mit PUR-Mantel)
zum Anschluss des PG und der ET 200S am PROFIBUS-DP-Netz	Busanschlussstecker ohne PG-Buchse (bis 12 MBaud)	6ES7 972-0BA10-0XA0 (mit geradem Kabelabgang) 6ES7 972-0BA40-0XA0 (mit schrägem Kabelabgang)
zum Doppelanschluss, z. B. von PG und DP-Master am PROFIBUS-DP-Netz über eine DP-Schnittstelle (siehe Bild 4-5)	Busanschlussstecker mit PG-Buchse (bis 12 MBaud)	6ES7 972-0BB10-0XA0 (mit geradem Kabelabgang) 6ES7 972-0BB40-0XA0 (mit schrägem Kabelabgang)
zum Anschluss des PG am Busanschlussstecker mit PG-Buchse	PG-Steckleitung (bis 1,5 MBaud)	6ES7 901-4BD00-0XA0

Beispiel für die Verwendung von Netzkomponenten

Im folgenden Bild ist das Beispiel aus Bild 4-3 mit Verwendung der Netzkomponenten dargestellt. Der Anschluss des Buskabels an die Busanschlussstecker ist in der Produktinformation zum jeweiligen Busanschlussstecker beschrieben.

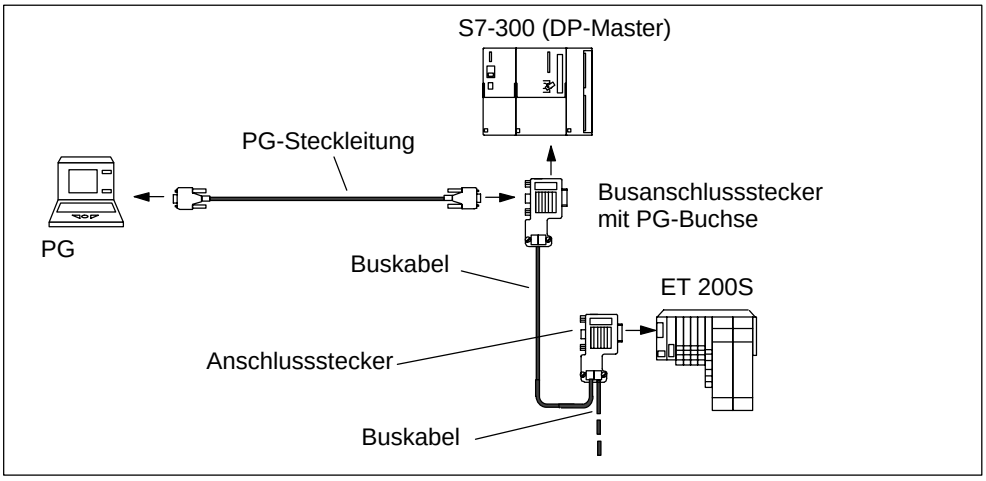


Bild 4-5 DP-Netz anschließen

Anschluss des IM 151-7 CPU FO

Informationen zum Anschluss und zur Verdrahtung der Lichtwellenleiter erhalten Sie im Handbuch *Dezentales Peripheriegerät ET 200S Verdrahten und Bestücken.*

4.3 PROFIBUS-Adresse

Eigenschaften

Mit der PROFIBUS-Adresse legen Sie fest, unter welcher Adresse das IM 151-7 CPU am PROFIBUS-DP angesprochen wird.

Voraussetzungen

- Erlaubte PROFIBUS-DP Adressen sind 1 bis 125.
- Jede Adresse darf nur einmal am PROFIBUS-DP vergeben werden.

Anlauf ohne DP-Projektierung auf der MMC (Erstanlauf)

Nach NETZ-EIN läuft die koexistente Schnittstelle auf dem IM 151-7 CPU als MPI-Schnittstelle mit Adresse 2, HSA 31 und 187,5 kBaud hoch. Die DP-Slave-Funktionalität des IM 151-7 CPU steht noch nicht zur Verfügung. Über die Schnittstelle sind alle im Kapitel 4.4 aufgelisteten PG-Funktionen möglich.

Mehrere ET 200S mit IM 151-7 CPU als DP-Slaves an einem PROFIBUS-Netz müssen schrittweise in Betrieb genommen werden. Nach dem Einschalten jedes einzelnen IM 151-7 CPU muss mit STEP 7 eine Projektierung mit DP-Adresse auf das IM 151-7 CPU übertragen werden.

Hinweis

Die Busparameter sind remanent. D. h., einmal projektierte Busparameter (z. B. Adresse, Baudrate) bleiben auch erhalten

- über NETZ-AUS
- wenn keine Projektierung mehr auf dem IM 151-7 CPU vorhanden ist (z. B. nach Löschen der SDBs, NETZ-EIN ohne MMC)

Anlauf mit DP-Projektierung auf der MMC

Sobald einmal eine DP-Projektierung in das IM 151-7 CPU geladen wurde, werden beim Anlauf die auf der MMC gespeicherten Daten verwendet.

Als DP-Slave läuft das IM 151-7 CPU nach NETZ-EIN mit der projektierten Adresse hoch und wartet auf die Parametrierung durch den DP-Master.

Als aktiver PROFIBUS-Teilnehmer übernimmt das IM 151-7 CPU die projektierte Baudrate.
Als passiver PROFIBUS-Teilnehmer sucht das IM 151-7 CPU die Baudrate.

4.4 Funktionen über PG/OP

Über das PG können Sie:

- IM 151-7 CPU mit ET 200S-Modulen projektieren und am PROFIBUS-DP in Betrieb nehmen.
- das IM 151-7 CPU programmieren.
- Testfunktionen wie “Variable beobachten und steuern” und “Programmstatus” ausführen.

  Inbetriebnahmefunktionen wie “Starten” und “Urlöschen” ausführen.
- den “Baugruppenzustand anzeigen” lassen; d. h. für das IM 151-7 CPU z. B. Auslastung des Lade- und Arbeitsspeichers, Stack-Inhalte und Diagnosepuffer-Inhalt anzeigen lassen.

Über das OP können Sie:

- Bedienen und Beobachten.

Eine ausführliche Beschreibung der Funktionen finden Sie in der Online-Hilfe zu *STEP 7*.

IM 151-7 CPU als passiven DP-Slave am PG betreiben – notwendige Einstellungen in *STEP 7*

Wenn Sie ein IM 151-7 CPU direkt an einem PG anschließen, dann müssen Sie für die Kommunikation zwischen den beiden Partnern eine Einstellung der PG-Schnittstelle in *STEP 7* vornehmen. Gehen Sie wie folgt vor:

1. Rufen Sie in *STEP 7* das Werkzeug “PG-/PC-Schnittstelle einstellen” auf (von der Windows-Startleiste aus **Start > STEP 7 > PG-PC-Schnittstelle einstellen** anwählen).
2. Stellen Sie die Schnittstelle Ihres PG auf PROFIBUS ein.
3. Rufen Sie die Eigenschaften des PROFIBUS-Netzes auf.
4. Stellen Sie die Eigenschaften so ein, dass das PG/PC als “einziger Master am Bus” aktiv ist.

Wenn Sie später einen DP-Master für das Netz projektiert haben und online gehen wollen, dann sollten Sie diese Einstellungen wieder zurückstellen, da dadurch zusätzliche Sicherheitsfunktionen gegen Busstörungen aktiviert werden.

Testfunktion Forcen

Mit der Funktion “Forcen” können Sie beim IM 151-7 CPU Eingänge und Ausgänge im Prozessabbild mit festen Werten vorbelegen.

Diese von Ihnen vorbelegten Werte (Forcewerte) können beim IM 151-7 CPU vom Anwenderprogramm und durch PG/OP-Funktionen weiterhin beeinflusst werden. Dieses Prinzip zeigt Bild 4-6.

Sie können beim IM 151-7 CPU bis zu maximal 10 Variablen forcen.

Vorsicht

Die Forcewerte im Prozessabbild der Eingänge können durch schreibende Befehle (zum Beispiel T EB x, = E x.y, Kopieren mit SFC usw.) und durch lesende Peripheriebefehle (zum Beispiel L PEW x) im Anwenderprogramm oder auch durch schreibende PG/OP-Funktionen überschrieben werden!

Mit Forcewerten vorbelegte Ausgänge liefern nur dann den Forcewert, wenn im Anwenderprogramm nicht mit schreibenden Peripheriebefehlen (zum Beispiel T PAB x) auf die Ausgänge geschrieben wird und keine PG/OP-Funktionen auf diese Ausgänge schreiben!

Achten Sie unbedingt darauf, dass Forcewerte im Prozessabbild der Ein-/Ausgänge nicht durch das Anwenderprogramm bzw. durch PG/OP-Funktionen überschrieben werden können!

Prinzip des Forcens beim IM 151-7 CPU

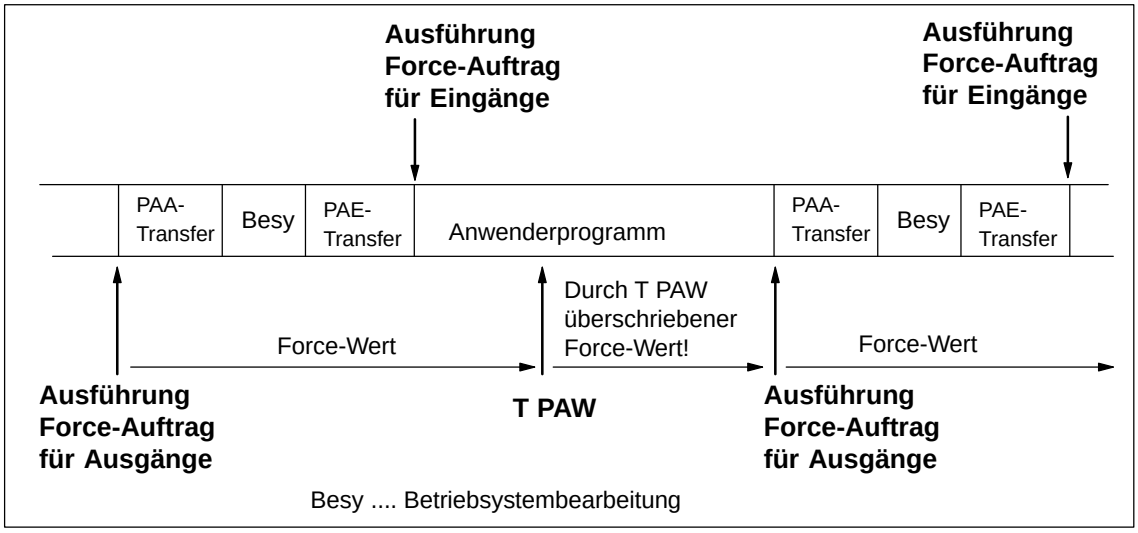


Bild 4-6 Prinzip des Forcens

Anwendungsbeispiel

Voraussetzung: In Ihrem Anwenderprogramm erfolgt kein Peripheriedirektzugriff.

Wenn in Ihrer Anlage beispielsweise ein Freigabesensor f defekt ist und Ihrem Anwenderprogramm z. B. über den Eingang 1.2 dauerhaft eine logische 0 signalisiert, können Sie diesen Sensor quasi überbrücken, indem Sie den Eingang auf 1 forcen, um den Betrieb Ihrer Anlage aufrecht zu erhalten.

Warnung

Da der Sensor außer Betrieb ist, müssen Sie die Funktionalität auf andere Weise überwachen, um Schaden für Bediener oder Maschine zu vermeiden.

4.5 Direkter Datenaustausch

Sie können mit *STEP 7* V5.1 das IM 151-7 CPU als I-Slave für direkten Datenaustausch projektieren. Direkter Datenaustausch ist eine spezielle Kommunikationsbeziehung zwischen PROFIBUS-DP-Teilnehmern.

Prinzip

Der direkte Datenaustausch ist dadurch gekennzeichnet, dass PROFIBUS-DP-Teilnehmer “mithören”, welche Daten ein DP-Slave seinem DP-Master zurückschickt. Durch diesen Mechanismus kann der “Mithörer” (Empfänger) direkt auf Änderungen von Eingangsdaten entfernter DP-Slaves zugreifen.

Bei der Projektierung in *STEP 7* legen Sie über die jeweiligen Peripherieeingangsadressen fest, auf welchen Adressbereich des Empfängers die gewünschten Daten des Senders gelesen werden sollen.

Beispiel

Das Bild 4-7 zeigt an einem Beispiel, welche Beziehungen Sie in *STEP 7* V5.1 für direkten Datenaustausch mit einem IM 151-7 CPU projektieren können. Andere DP-Slaves können hier nur Sender sein.

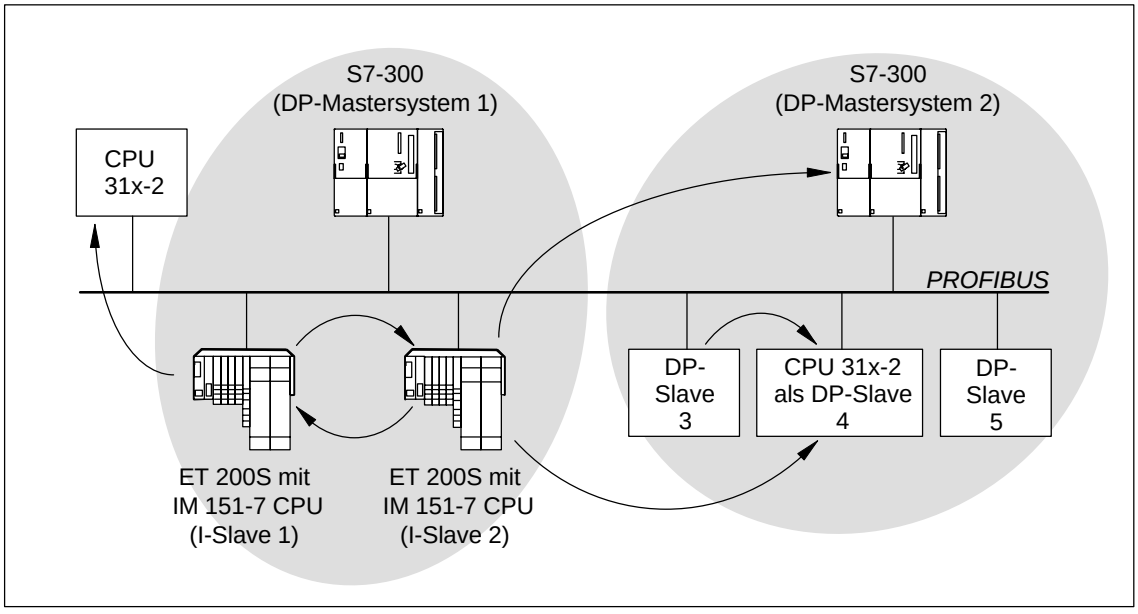


Bild 4-7 Direkter Datenaustausch mit IM 151-7 CPU

Funktionalität im direkten Datenaustausch

Das IM 151-7 CPU besitzt folgende Möglichkeiten im direkten Datenaustausch:

- Sender:
  Das IM 151-7 CPU sendet als DP-Slave die für den direkten Datenaustausch projektierten Prozessausgänge als Broadcast-Telegramm an alle Busteilnehmer. Aus diesem Broadcast-Telegramm filtern sich andere Empfänger die relevanten Daten heraus.
- Empfänger:
  Herausfiltern der Daten aus dem Broadcast-Telegramm von Sendern, die über *STEP 7* für den direkten Datenaustausch als relevant projektiert wurden.

Diagnose bei direktem Datenaustausch

Zur Diagnose der für den direkten Datenaustausch projektierten DP-Slaves können nur die Ergebnisse der Verbindungsüberwachung herangezogen werden, da Diagnosemeldungen der “abgehörten” DP-Slaves nur dem DP-Master gemeldet werden.

Bei Stationsausfall und -wiederkehr wird der Asynchron-OB 86 aufgerufen. Bei Zugriff auf Daten während eines Stationsausfalles des Senders wird ein Peripheriezugriffsfehler erkannt und OB 122 aufgerufen. Für die Baugruppenzustandsdaten sind nur die Kennungen “Baugruppe vorhanden” und “Baugruppe verfügbar” relevant.

ET 200S im MPI-Netz

5

Einleitung

Sie können ET 200S mit IM 151-7 CPU als Teilnehmer in ein MPI-Netz integrieren. In diesem Kapitel finden Sie einen typischen Netz-Aufbau mit IM 151-7 CPU erläutert. Welche Funktionen über ein PG oder OP auf IM 151-7 CPU ausführbar sind, finden Sie im Kapitel 4.4. Die verfügbaren Kommunikationsdienste finden Sie im Kapitel 7.7.

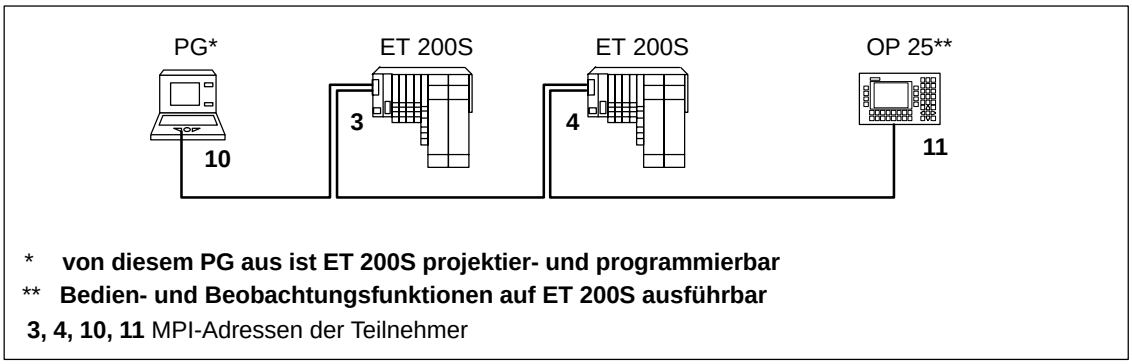
Kapitelübersicht

Kapitel	Thema	Seite
5.1	ET 200S im MPI-Netz	5-2
5.2	MPI-Adresse	5-3

5.1 ET 200S im MPI-Netz

Aufbau eines MPI-Netzes

Das folgende Bild zeigt ein Beispiel für den Aufbau eines MPI-Netzes.



* **von diesem PG aus ist ET 200S projektier- und programmierbar**
** **Bedien- und Beobachtungsfunktionen auf ET 200S ausführbar**
3, 4, 10, 11 MPI-Adressen der Teilnehmer

Bild 5-1 Beispiel für ein MPI-Netz

Hardware-Voraussetzung im PG/OP für Zugriff auf ET 200S

Damit Sie von einem PG/OP aus auf ein IM 151-7 CPU zugreifen können, muss das PG/OP folgende Voraussetzungen erfüllen:

- mit einer integrierten MPI-Schnittstelle oder MPI-Karte ausgestattet sein oder
- mit einer integrierten PROFIBUS-DP-Schnittstelle oder DP-Karte ausgestattet sein

Baudraten

Im MPI-Netz sind mit dem IM 151-7 CPU alle MPI-Baudraten möglich.

Netzkomponenten

Für den Aufbau eines MPI-Netzes verwenden Sie die gleichen Netzkomponenten wie für ein PROFIBUS-DP-Netz (siehe Kapitel 4.2).

5.2 MPI-Adresse

Eigenschaften

Mit der MPI-Adresse legen Sie fest, unter welcher Adresse das IM 151-7 CPU im MPI-Netz angesprochen wird.

Voraussetzungen

- Erlaubte MPI-Adressen sind 0 bis 126.
- Jede Adresse darf nur einmal am MPI-Netz vergeben werden.

Empfehlung für MPI-Adressen

- Vergeben Sie MPI-Adressen größer “2” an die festen Teilnehmer im MPI-Netz.
- Reservieren Sie die MPI-Adresse “0” für ein Service-PG bzw. “1” für ein Service-OP, die später bei Bedarf kurzzeitig an das MPI-Netz angeschlossen werden.
- Reservieren Sie die MPI-Adresse “2” für eine CPU. So vermeiden Sie das Auftreten von doppelten MPI-Adressen nach Einbau einer CPU mit Defaulteinstellung in das MPI-Netz (z. B. beim Austausch einer CPU).

Anlauf ohne Projektierung auf der MMC (Erstanlauf)

Nach NETZ-EIN läuft die koexistente Schnittstelle auf dem IM 151-7 CPU als MPI-Schnittstelle mit der Adresse 2, HSA 31 und 187,5 kBaud hoch. Über die Schnittstelle sind alle im Kapitel 4.4 aufgelisteten PG-Funktionen möglich.

Hinweis

Die Busparameter sind remanent. D. h., einmal projektierte Busparameter (z. B. Adresse, Baudrate) bleiben auch erhalten

- über NETZ-AUS
- wenn keine Projektierung mehr auf dem IM 151-7 CPU vorhanden ist (z. B. nach Löschen der SDBs, NETZ-EIN ohne MMC)

Anlauf mit Projektierung auf der MMC

Sobald einmal eine Projektierung in das IM 151-7 CPU geladen wurde, werden beim Anlauf die auf der MMC gespeicherten Daten verwendet.

Inbetriebnahme und Diagnose

6

IM 151-7 CPU mit *STEP 7* projektieren

Sie finden in diesem Kapitel das Projektieren einer ET 200S für IM 151-7 CPU mit *STEP 7* beschrieben.

IM 151-7 CPU urlöschen

Unter bestimmten Bedingungen müssen Sie das IM 151-7 CPU urlöschen. In diesem Kapitel finden Sie diese Bedingungen und die Vorgehensweise zum Urlöschen beschrieben.

Diagnosemöglichkeiten

Das Dezentrale Peripheriesystem ET 200S ist auf eine möglichst einfache Handhabung und Inbetriebnahme ausgelegt. Falls trotzdem ein Fehler auftreten sollte, können Sie diesen Fehler mit Hilfe der LED-Anzeige, der Slave-Diagnose und der Diagnosemöglichkeiten in *STEP 7* auswerten.

Alarmauswertung

Zur Auswertung der **Alarme** durch ET 200S zeigen wir Ihnen den Unterschied hinsichtlich der Alarmmeldung zum S7/M7-DP-Master und zu anderen DP-Mastern.

Kapitelübersicht

Kapitel	Thema	Seite
6.1	IM 151-7 CPU projektieren	6-2
6.2	IM 151-7 CPU urlöschen	6-4
6.3	Inbetriebnahme und Anlauf von ET 200S	6-7
6.4	Diagnose durch LED-Anzeige	6-9
6.5	Diagnose über Diagnoseadresse mit *STEP 7*	6-11
6.6	Slave-Diagnose bei Einsatz des IM 151-7 CPU als I-Slave	6-14
6.7	Diagnosedaten der Elektronikmodule	6-25

6.1 IM 151-7 CPU projektieren

Das Interfacemodul IM 151-7 CPU projektieren Sie als einen DP-Slave oder als Stand-alone-Baugruppe (MPI).

Das IM 151-7 CPU stellt sich dem Anwender in *STEP 7* als S7-300-Baugruppe dar, welche immer zusammen mit einem Rack in einer S7-300-Station erstellt wird. Ebenso kann die Baugruppe nur mit dem Rack gelöscht werden!

In einer S7-300-Station, welche ein IM 151-7 CPU beinhaltet, sind keine Erweiterungsracks projektierbar. Das IM 151-7 CPU wird dabei auf Steckplatz 2 angelegt und erhält ein MPI/DP-Submodul. Diese Projektierung gilt sowohl für die Variante mit RS 485 als auch mit LWL-Anbindung. Die ersten steckbaren Module sind ab Steckplatz 4 projektierbar.

Folgende Projektiermöglichkeiten ergeben sich:

Tabelle 6-1 Projektiermöglichkeiten

Projektierumgebung	Projektierwerkzeug	Projektierbare Betriebsart
SIMATIC S7	*STEP 7* (HW Konfig) ab V5.1 + Service Pack 4	• Stand-alone (MPI) • IM 151-7 CPU als S7-Slave
SIMATIC S5	COM PROFIBUS	vollständig projektiertes und programmiertes IM 151-7 CPU, eingebracht als Norm-I-Slave in COM PROFIBUS
Fremdsysteme	Fremd-Tool	vollständig projektiertes und programmiertes IM 151-7 CPU, eingebracht als Norm-I-Slave in Fremd-Tool

Voraussetzung

Sie haben *STEP 7* (ab V5.1 + Service Pack 4) geöffnet und befinden sich im SIMATIC Manager von *STEP 7*.

Vorgehensweise

Gehen Sie folgendermaßen vor:

1. Projektieren Sie das IM 151-7 CPU als eine S7-300-Station.
   - Legen Sie eine neue Station vom Stationstyp **S7-300** an (Menübefehl **Einfügen → Station**).
   - Wechseln Sie für diese Station in das Fenster zur Hardwarekonfiguration.
   - Wählen Sie im Fenster “Hardware Katalog” das Verzeichnis PROFIBUS-DP/ ET 200S/IM 151-7 CPU.
   - Ziehen Sie das Objekt “IM 151-7 CPU” per Drag & Drop in das leere Stationsfenster.
   - Konfigurieren Sie das ET 200S mit den gewünschten Peripheriemodulen.
   - Speichern Sie die Station (d. h. das ET 200S).
2. Projektieren Sie in einer anderen Station im gleichen Projekt einen DP-Master (z. B. CPU mit integrierter PROFIBUS-DP Schnittstelle oder CP 342-5 mit PROFIBUS-DP Schnittstelle ab 6GK7 342-5DA01-0XE0, Erzeugnisstand 2).
3. Ziehen Sie das ET 200S (mit IM 151-7 CPU) aus dem Fenster “Hardware Katalog” (Behälter **bereits projektierte Stationen**) per Drag & Drop auf das Symbol für das DP-Mastersystem.
4. Doppelklicken Sie auf das Symbol für den intelligenten DP-Slave und wählen Sie das Register “Kopplung”. In diesem Register treffen Sie die Zuordnung, welche Station hier den intelligenten DP-Slave repräsentieren soll.
5. Markieren Sie den intelligenten DP-Slave und klicken Sie auf die Schaltfläche “Koppeln”.
6. Wählen Sie das Register (Slave-) Konfiguration und ordnen Sie Master- und Slaveadressen einander zu.
7. Bestätigen Sie die Einstellungen mit “OK”.
8. Beide Stationen müssen anschließend neu geladen werden, um die Master-Slave-Kommunikation in Betrieb zu nehmen.

Projektierung in einem Fremdsystem

Mit Hilfe der GSD-Datei können Sie das IM 151-7 CPU auch in Fremdsysteme als DP-Norm-Slave einbinden. In diesem Fall besteht das Diagnosetelegramm aus:

- Stationsstatus
- Master-PROFIBUS-Adresse
- Herstellerkennung
- Kennungsbezogene Diagnose
- Modulstatus

6.2 IM 151-7 CPU urlöschen

Wann IM 151-7 CPU urlöschen?

Das IM 151-7 CPU muss urgelöscht werden,

- um remanente Bereiche zu löschen (Merker, Zeiten, Zähler).
- wenn das IM 151-7 CPU durch Blinken der STOP-LED mit 0,5 Hz das Urlöschen anfordert.

Mögliche Ursachen für die Urlöschanforderung können sein:

- erstmaliger Anlauf des ET 200S
- inkonsistente Speicherbereiche
- Speichermodul (MMC) wurde getauscht

Wie urlöschen?

Es gibt zwei Möglichkeiten, IM 151-7 CPU urzulöschen:

Tabelle 6-2 Möglichkeiten zum Urlöschen

Urlöschen mit Betriebsartenschalter	Urlöschen mit PG
... wird in diesem Kapitel beschrieben	... ist nur möglich im STOP der CPU (siehe PG-Handbücher und *STEP 7 Online-Hilfe*)

IM 151-7 CPU urlöschen mit Betriebsartenschalter

Um IM 151-7 CPU mit Hilfe des Betriebsartenschalters urzulöschen, gehen Sie wie folgt vor (siehe auch Bild 6-1):

1. Schalten Sie den Betriebsartenschalter in Stellung STOP.
2. Drücken Sie den Betriebsartenschalter in Stellung MRES. Halten Sie den Schalter in dieser Stellung, bis die STOP-LED zum 2. Mal aufleuchtet (entspricht 3 Sekunden) und lassen Sie ihn dann wieder in die Stellung STOP schnappen.
3. Innerhalb von 3 Sekunden müssen Sie den Betriebsartenschalter wieder in die Stellung MRES drücken und solange in dieser Stellung halten, bis die STOP-LED schnell blinkt (mit 2 Hz). Wenn IM 151-7 CPU das Urlöschen beendet hat, hört die STOP-LED auf zu blinken und leuchtet.

IM 151-7 CPU hat das Urlöschen durchgeführt.

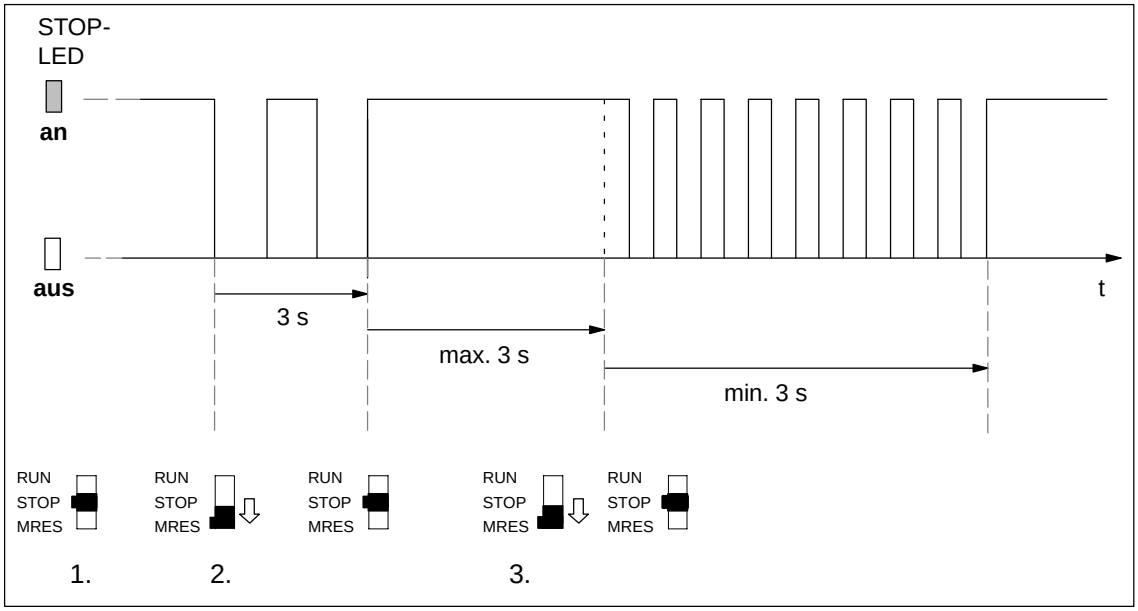


Bild 6-1 Bedienfolge des Betriebsartenschalters zum Urlöschen

STOP-LED blinkt nicht beim Urlöschen?

Die STOP-LED blinkt nicht beim Urlöschen oder andere Anzeigen leuchten? Dann müssen Sie die Schritte 2 und 3 wiederholen. Führt IM 151-7 CPU das Urlöschen wieder nicht durch, dann müssen Sie den Diagnosepuffer des CPU-Teils mit dem PG auslesen (siehe *Benutzerhandbuch STEP 7*).

Was passiert im CPU-Teil des IM 151-7 CPU?

Tabelle 6-3 CPU-interne Vorgänge beim Urlöschen

Vorgang	Verhalten der CPU im IM 151-7 CPU
Ablauf im CPU-Teil des IM 151-7 CPU	1. Die CPU löscht das gesamte Anwenderprogramm im Arbeitsspeicher und den RAM-Ladespeicher. 2. Die CPU löscht die remanenten Daten. 3. Die CPU testet ihre Hardware. 4. Wenn ein Speichermodul (Micro Memory Card = MMC) gesteckt ist, kopiert die CPU den ablaufrelevanten Inhalt des Moduls in den Arbeitsspeicher.
Speicherinhalte nach dem Urlöschen	Die CPU hat den Speicherfüllstand “0”. Wenn eine SIMATIC Micro Memory Card steckt, dann wird das Anwenderprogramm wieder in den Arbeitsspeicher übertragen.
Was bleibt erhalten?	Inhalt des Diagnosepuffers und des Betriebsstundenzählers

Hinweis

Wenn die CPU den Inhalt des Speichermoduls (MMC) nicht kopieren kann und Urlöschen anfordert:

- MMC ziehen
- CPU urlöschen
- Diagnosepuffer auslesen

Den Diagnosepuffer können Sie mit dem PG auslesen (siehe *STEP 7 Online-Hilfe*).

6.3 Inbetriebnahme und Anlauf von ET 200S

ET 200S in Betrieb nehmen

Nehmen Sie das Dezentrale Peripheriesystem ET 200S wie folgt in Betrieb:

1. Montieren Sie das Dezentrale Peripheriesystem ET 200S (siehe Handbuch *Dezentrales Peripheriesystem ET 200S*).
2. Verdrahten Sie das Dezentrale Peripheriesystem ET 200S (siehe Handbuch *Dezentrales Peripheriesystem ET 200S*).
3. Legen Sie bei Projektierung als DP-Slave in der Projektiersoftware die Adressbereiche im IM 151-7 CPU fest, über die der Datenaustausch mit dem DP-Master erfolgt (oder Defaulteinstellung für ET 200S verwenden; siehe Kapitel 3.4).
4. Schalten Sie die Geberversorgungsspannung für das ET 200S ein.
5. Schalten Sie ggf. die Lastspannung und die Versorgungsspannung für die Motorstarter ein.
6. Schalten Sie ggf. den CPU-Teil des IM 151-7 CPU in den Betriebszustand STOP.
7. Laden Sie die Projektierung für IM 151-7 CPU in das ET 200S.
8. Schalten Sie das IM 151-7 CPU in den Betriebszustand RUN.

Anlaufverhalten des IM 151-7 CPU

Achten Sie beim Anlauf des IM 151-7 CPU nach NETZ-EIN darauf, dass

- das Abschlussmodul gesteckt ist.
- alle mit dem IM 151-7 CPU verbundenen Terminalmodule bestückt sind.

Andernfalls bleibt das IM 151-7 CPU im Betriebszustand ANLAUF.

Anwenderprogramm laden

Bei der Inbetriebnahme von ET 200S haben Sie folgende Möglichkeiten, das Anwenderprogramm in das IM 151-7 CPU zu laden:

- Das Programm wird vom PG/PC über “Anwenderprogramm laden” auf das im IM 151-7 CPU steckende Speichermodul (MMC) geladen.

Hinweis

Bei dieser Funktion werden remanente Bereiche nicht gelöscht.

- Das Programm wird am PG/PC auf das Speichermodul (MMC) übertragen. Danach wird das Speichermodul in das IM 151-7 CPU gesteckt und die Urlöschanforderung quittiert.

Siehe dazu Kapitel 7.3.

Tipp: OB 82 und 86 bei der Inbetriebnahme programmieren

Programmieren Sie bei der Inbetriebnahme als DP-Slave mit *STEP 7* im DP-Master und im DP-Slave immer die OBs 82 und 86. So können Sie die jeweiligen Betriebszustände bzw. Unterbrechungen des Nutzdatentransfers erkennen und auswerten (siehe Tabellen 6-5 und 6-6).

Hinweis

Ohne Projektierung ist ein Default-Anlauf möglich, wenn die Powermodule eingeschaltet sind und alle Module stecken.

6.4 Diagnose durch LED-Anzeige

LED-Anzeige

Die LEDs RUN, STOP, ON, BF, SF und FRCE dienen der Signalisierung der für den Anwender wichtigen Zustandsinformationen des IM 151-7 CPU.

Das IM 151-7 CPU verfügt über die folgenden 6 LEDs:

- LED “SF” (**S**ystem **F**ault) zur Anzeige eines Fehlers im ET 200S
- LED “BF” (**B**us **F**ault) zur Anzeige von Fehlern am PROFIBUS-DP
- LED “ON” leuchtet, wenn ET 200S an eine Versorgungsspannung angeschlossen ist
- LED “FRCE” leuchtet, wenn ein Force-Auftrag aktiv ist
- LED “RUN” leuchtet, wenn der CPU-Teil des IM 151-7 CPU im Betriebszustand RUN ist
- LED “STOP” leuchtet, wenn der CPU-Teil des IM 151-7 CPU im Betriebszustand STOP ist

Die Bedeutung der LEDs für die CPU-Funtionalität finden Sie im Kapitel 7.2 ausführlich beschrieben.

LED “ON” leuchtet nicht

Wenn die LED “ON” nicht leuchtet, dann liegt entweder keine oder eine zu geringe Versorgungsspannung für Elektronik/Geber am ET 200S an. Mögliche Ursachen sind eine defekte Sicherung oder fehlende bzw. zu geringe Netzspannung.

Diagnose der DP-Funktionalität mit Hilfe der LEDs “BF” und “SF”

Wenn die LEDs “BF” und “SF” leuchten oder blinken, dann ist die Projektierung des ET 200S nicht in Ordnung. In der folgenden Tabelle finden Sie mögliche Fehleranzeigen mit ihrer Bedeutung und Abhilfe.

In der folgenden Tabelle sind die LED-Zustände für den DP-Slave-Betrieb dargestellt. Im Stand-alone-Betrieb (MPI) ist die DP-Funktionalität irrelevant und es wird keine BF-LED angesteuert (keine LED-Anzeige für die Baudratensuche).

Tabelle 6-4 LED-Anzeige für PROFIBUS-DP

LED “BF”	LED “SF”	Bedeutung	Ursache	Fehlerbehandlung
ein	ein	keine Verbindung zu DP-Master	• IM 151-7 CPU ist aktiver Busteilnehmer ⇒ Buskurzschluss • IM 151-7 CPU ist passiver Busteilnehmer ⇒ Baudratensuche: kein aktiver Teilnehmer am Bus, DP-Master nicht vorhanden oder ausgeschaltet oder Busanschluss unterbrochen SF ist an wegen Stationsausfall	• Überprüfen Sie, ob der Anschlussstecker für PROFIBUS-DP richtig steckt • Überprüfen Sie, ob das Buskabel zum DP-Master defekt ist
blinkt	ein	Parametrierfehler, es findet kein Datenaustausch statt	• Slave ist nicht oder falsch projektiert • falsche aber zulässige Stationsadresse projektiert • projektierte Adressbereiche von Istaufbau stimmen nicht mit Sollaufbau überein • Stationsaufall eines parametrierten Senders im direkten Datenaustausch • DP-Master nicht vorhanden oder ausgeschaltet	• Überprüfen Sie die Hardware des ET 200S • Überprüfen Sie die Konfigurierung und Parametrierung des ET 200S • Überprüfen Sie die Einstellung der projektierten Adressbereiche für den Master
aus	ein	Fehler im Slave: Diagnosealarm	Master in STOP-Zustand	Schalten Sie den DP-Master in den Betriebszustand RUN
aus	aus	Es findet ein Datenaustausch statt	Soll- und Ist-Konfiguration der ET 200S stimmen überein	

6.5 Diagnose über Diagnoseadresse mit *STEP 7*

Aufgetretene Fehler im ET 200S werden mit der LED “SF” angezeigt und die Fehlerursache wird in den Diagnosepuffer des IM 151-7 CPU eingetragen. Der CPU-Teil des IM 151-7 CPU geht entweder in STOP oder Sie können im Anwenderprogramm über Fehler- bzw. Alarm-OBs auf Fehler reagieren.

Damit eine Reaktion möglich ist, muss der Verursacher über eine Diagnoseadresse identifiziert werden können.

Diagnoseadressen

Wenn Sie das ET 200S mit einem DP-Master aus dem SIMATIC S7-Spektrum am PROFIBUS-DP betreiben, dann werden in *STEP 7* Diagnoseadressen wie folgt vergeben:

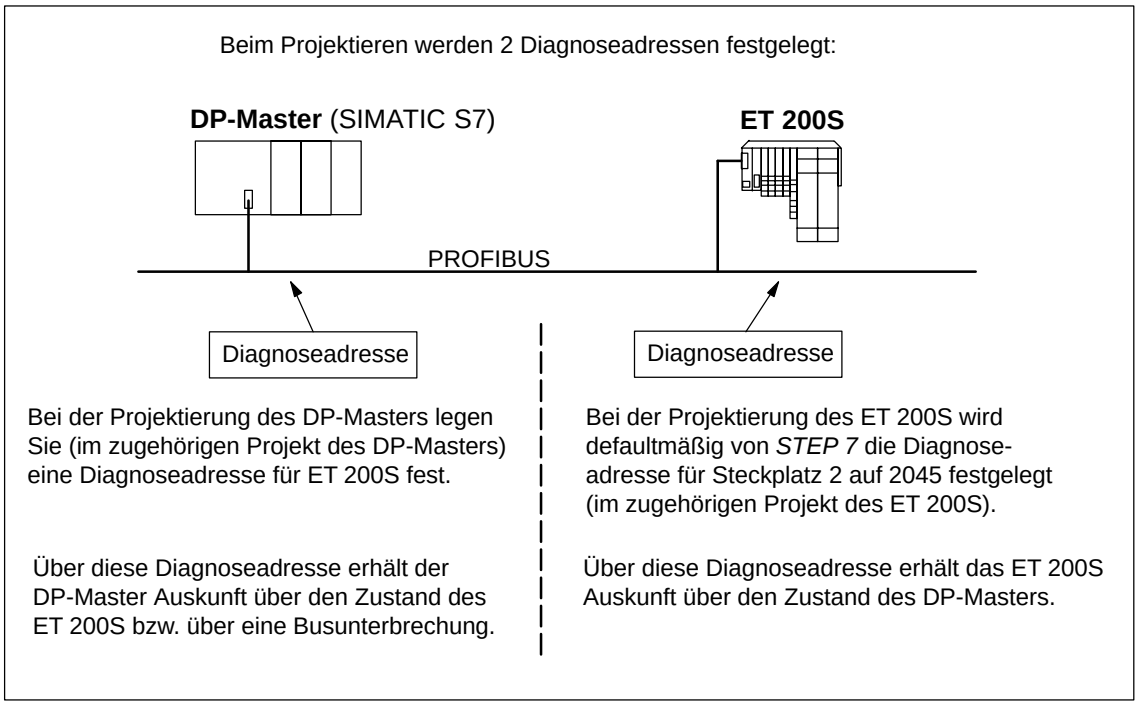

Beim Projektieren werden 2 Diagnoseadressen festgelegt:
DP-Master (SIMATIC S7)
ET 200S
PROFIBUS
Diagnoseadresse
Diagnoseadresse
Bei der Projektierung des DP-Masters legen Sie (im zugehörigen Projekt des DP-Masters) eine Diagnoseadresse für ET 200S fest.
Über diese Diagnoseadresse erhält der DP-Master Auskunft über den Zustand des ET 200S bzw. über eine Busunterbrechung.
Bei der Projektierung des ET 200S wird defaultmäßig von STEP 7 die Diagnoseadresse für Steckplatz 2 auf 2045 festgelegt (im zugehörigen Projekt des ET 200S).
Über diese Diagnoseadresse erhält das ET 200S Auskunft über den Zustand des DP-Masters.

Bild 6-2 Diagnoseadressen für DP-Master und ET 200S

Ereigniserkennung

Die folgende Tabelle zeigt, wie der DP-Master bzw. das IM 151-7 CPU des ET 200S Betriebszustandsänderungen bzw. Unterbrechungen des Nutzdatentransfers erkennt.

Tabelle 6-5 Reaktionen auf Betriebszustandsänderungen bzw. Unterbrechungen des Nutzdatentransfers im DP-Master und im ET 200S mit IM 151-7 CPU

Ereignis	was passiert ...	
	im DP-Master	**im IM 151-7 CPU**
Busunterbrechung (Kurzschluss, Stecker gezogen)	• Aufruf des OB 86 mit der Meldung *Stationsausfall* (kommendes Ereignis; Diagnoseadresse des IM 151-7 CPU) • bei Peripheriezugriff auf Übergabebereich: Aufruf des OB 122 (Peripheriezugriffsfehler)	• Aufruf des OB 86 mit der Meldung *Stationsausfall* (kommendes Ereignis; Diagnoseadresse des IM 151-7 CPU) • bei Peripheriezugriff auf Übergabebereich: Aufruf des OB 122 (Peripheriezugriffsfehler)
ET 200S: RUN → STOP	• Aufruf des OB 82 mit der Meldung *Baugruppe gestört* (kommendes Ereignis; Diagnoseadresse des IM 151-7 CPU; Variable OB82_MDL_STOP=1)	–
ET 200S: STOP → RUN	• Aufruf des OB 82 mit der Meldung *Baugruppe ok.* (gehendes Ereignis; Diagnoseadresse des IM 151-7 CPU; Variable OB82_MDL_STOP=0)	–
DP-Master: RUN → STOP	–	• Aufruf des OB 82 mit der Meldung *Baugruppe gestört* (kommendes Ereignis; Diagnoseadresse des IM 151-7 CPU; Variable OB82_MDL_STOP=1)
DP-Master: STOP → RUN	–	• Aufruf des OB 82 mit der Meldung *Baugruppe ok.* (gehendes Ereignis; Diagnoseadresse des IM 151-7 CPU; Variable OB82_MDL_STOP=0)

Auswertung im Anwenderprogramm

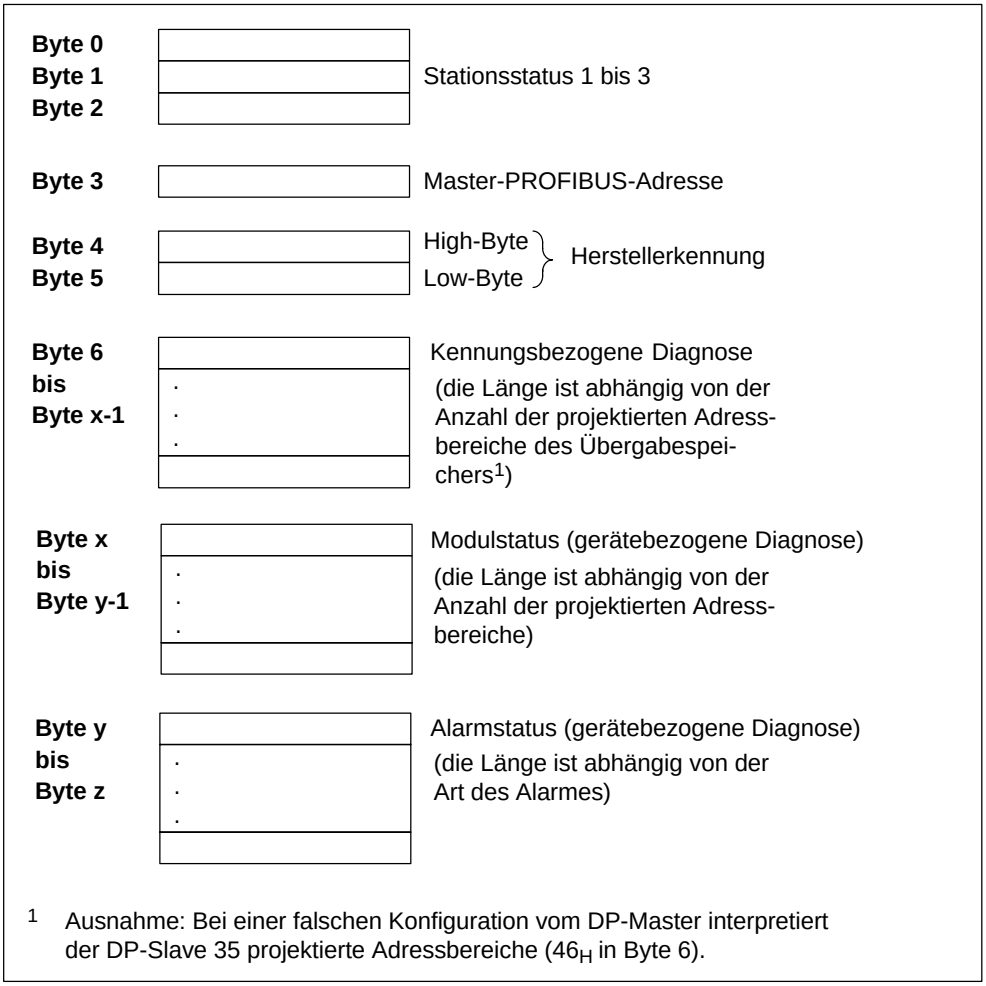
Die folgende Tabelle zeigt Ihnen, wie Sie zum Beispiel RUN-STOP-Übergänge im DP-Master (CPU 315-2 DP; 6ES7 315-2AF03-0AB0) bzw. im ET 200S auswerten können.

Tabelle 6-6 Auswertung von RUN-STOP-Übergängen im DP-Master/im ET 200S

im DP-Master		im ET 200S (IM 151-7 CPU)
Diagnoseadressen: (Beispiel) Master-Diagnoseadresse=1023 Slave-Diagnoseadresse im Mastersystem=**1022**		Diagnoseadressen: (Beispiel) Slave-Diagnoseadresse Steckplatz 2=**2045** Master-Diagnoseadresse=nicht relevant
Die CPU ruft den OB 82 auf mit u. a. folgenden Informationen: • OB82_MDL_ADDR:=**1022** • OB82_EV_CLASS:=B#16#39 (kommendes Ereignis) • OB82_MDL_DEFECT:=Baugruppenstörung Tipp: diese Informationen stehen auch im Diagnosepuffer der CPU Im Anwenderprogramm sollten Sie auch den SFC 13 “DPNRM_DG” zum Auslesen der Slave-Diagnose programmieren.	←	CPU in IM 151-7 CPU: RUN → STOP CPU erzeugt ein Diagnosetelegramm (Slave-Diagnose, siehe Handbuch *Dezentrales Peripheriesystem ET 200S*).
CPU: RUN → STOP	→	Der CPU-Teil in IM 151-7 CPU ruft den OB 82 auf mit u. a. folgenden Informationen: • OB82_MDL_ADDR:=**2045** • OB82_EV_CLASS:=B#16#39 (kommendes Ereignis) • OB82_MDL_DEFECT:=Baugruppenstörung Tipp: diese Informationen stehen auch im Diagnosepuffer des CPU-Teils

6.6 Slave-Diagnose bei Einsatz des IM 151-7 CPU als I-Slave

Aufbau des Diagnosetelegramms



1 Ausnahme: Bei einer falschen Konfiguration vom DP-Master interpretiert der DP-Slave 35 projektierte Adressbereiche (46_H in Byte 6).

Bild 6-3 Aufbau der Slave-Diagnose

6.6.1 Stationsstatus 1 bis 3

Definition

Der Stationsstatus 1 bis 3 gibt einen Überblick über den Zustand eines DP-Slaves.

Stationsstatus 1

Tabelle 6-7 Aufbau von Stationsstatus 1 (Byte 0)

Bit	Bedeutung	Abhilfe
0	1: DP-Slave kann von DP-Master **nicht** angesprochen werden.	• Richtige DP-Adresse für DP-Slave projektiert? • Busanschlussstecker angeschlossen? • Spannung am DP-Slave? • RS 485-Repeater richtig eingestellt? • Reset am DP-Slave durchführen
1	1: DP-Slave ist für Datenaustausch noch nicht bereit.	• Abwarten, da DP-Slave gerade im Hochlauf ist.
2	1: Die vom DP-Master an den DP-Slave gesendeten Konfigurationsdaten stimmen nicht mit dem Aufbau des DP-Slaves überein.	• Richtiger Stationstyp oder richtiger Aufbau des DP-Slaves in der Software eingegeben?
3	1: Diagnosealarm, erzeugt durch RUN-STOP-Übergang der CPU oder durch den SFB 75 0: Diagnosealarm, erzeugt durch STOP-RUN-Übergang der CPU oder durch den SFB 75	• Sie können die Diagnose auslesen.
4	1: Funktion wird nicht unterstützt, z. B. Ändern der DP-Adresse über Software	• Überprüfen Sie die Projektierung.
5	0: Das Bit ist immer “0”.	–
6	1: DP-Slave-Typ stimmt nicht mit der Software-Projektierung überein.	• Richtiger Stationstyp in der Software eingegeben? (Parametrierfehler)
7	1: DP-Slave ist von einem anderen DP-Master parametriert worden als dem DP-Master, der im Augenblick Zugriff auf den DP-Slave hat.	• Bit ist immer 1, wenn Sie z. B. gerade mit dem PG oder einem anderen DP-Master auf den DP-Slave zugreifen. Die DP-Adresse des Parametriermasters befindet sich im Diagnosebyte “Master-PROFIBUS-Adresse”.

Stationsstatus 2

Tabelle 6-8 Aufbau von Stationsstatus 2 (Byte 1)

Bit	Bedeutung
0	1: DP-Slave muss neu parametriert und konfiguriert werden.
1	1: Es liegt eine Diagnosemeldung vor. Der DP-Slave kann nicht weiterlaufen, solange der Fehler nicht behoben ist (statische Diagnosemeldung).
2	1: Bit ist immer auf “1”, wenn DP-Slave mit dieser DP-Adresse vorhanden ist.
3	1: Es ist bei diesem DP-Slave die Ansprechüberwachung aktiviert.
4	1: DP-Slave hat Steuerkommando “FREEZE” erhalten.
5	1: DP-Slave hat Steuerkommando “SYNC” erhalten.
6	0: Bit ist immer auf “0”.
7	1: DP-Slave ist deaktiviert, d. h. er ist aus der zyklischen Bearbeitung herausgenommen.

Stationsstatus 3

Tabelle 6-9 Aufbau von Stationsstatus 3 (Byte 2)

Bit	Bedeutung
0 bis 6	0: Bits sind immer auf “0”
7	1: • Es liegen mehr Diagnosemeldungen vor, als der DP-Slave speichern kann. • Der DP-Master kann nicht alle vom DP-Slave gesendeten Diagnosemeldungen in seinem Diagnosepuffer eintragen.

6.6.2 Master-PROFIBUS-Adresse

Definition

Im Diagnosebyte Master-PROFIBUS-Adresse ist die DP-Adresse des DP-Masters hinterlegt:

- der den DP-Slave parametriert hat und
- der lesenden und schreibenden Zugriff auf den DP-Slave hat

Master-PROFIBUS-Adresse

Tabelle 6-10 Aufbau der Master-PROFIBUS-Adresse (Byte 3)

Bit	Bedeutung
0 bis 7	DP-Adresse des DP-Masters, der den DP-Slave parametriert hat und lesenden und schreibenden Zugriff auf den DP-Slave hat.
	FF_H: DP-Slave wurde von keinem DP-Master parametriert.

6.6.3 Herstellerkennung

Definition

In der Herstellerkennung ist ein Code hinterlegt, der den Typ des DP-Slaves beschreibt.

Herstellerkennung

Tabelle 6-11 Aufbau der Herstellerkennung (Byte 4, 5)

Byte 4	Byte 5	Herstellerkennung für
80_H	$E2_H$	IM 151-7 CPU
80_H	$.._H$	IM 151-7 CPU FO

6.6.4 Kennungsbezogene Diagnose

Definition

Die kennungsbezogene Diagnose sagt aus, für welchen der projektierten Adressbereiche des Übergabespeichers ein Eintrag erfolgt ist.

Aufbau

Das folgende Bild zeigt den Aufbau der kennungsbezogenen Diagnose für die Maximalanzahl der projektierten Adressbereiche.

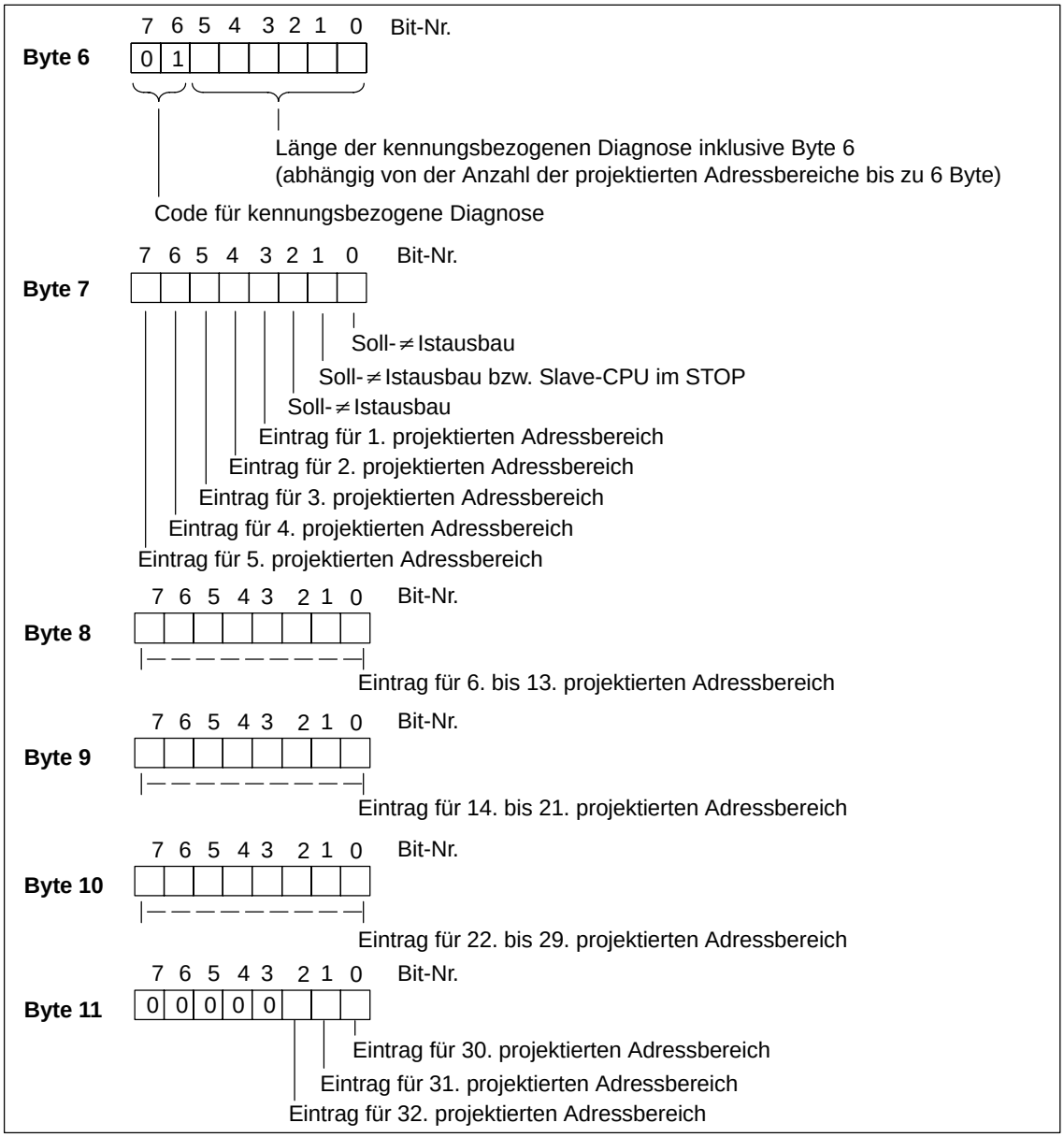

Byte 6
7 6 5 4 3 2 1 0 Bit-Nr.
0 1
Länge der kennungsbezogenen Diagnose inklusive Byte 6
(abhängig von der Anzahl der projektierten Adressbereiche bis zu 6 Byte)
Code für kennungsbezogene Diagnose
Byte 7
7 6 5 4 3 2 1 0 Bit-Nr.
Soll- ≠ Istausbau
Soll- ≠ Istausbau bzw. Slave-CPU im STOP
Soll- ≠ Istausbau
Eintrag für 1. projektierten Adressbereich
Eintrag für 2. projektierten Adressbereich
Eintrag für 3. projektierten Adressbereich
Eintrag für 4. projektierten Adressbereich
Eintrag für 5. projektierten Adressbereich
Byte 8
7 6 5 4 3 2 1 0 Bit-Nr.
Eintrag für 6. bis 13. projektierten Adressbereich
Byte 9
7 6 5 4 3 2 1 0 Bit-Nr.
Eintrag für 14. bis 21. projektierten Adressbereich
Byte 10
7 6 5 4 3 2 1 0 Bit-Nr.
Eintrag für 22. bis 29. projektierten Adressbereich
Byte 11
7 6 5 4 3 2 1 0 Bit-Nr.
0 0 0 0 0
Eintrag für 30. projektierten Adressbereich
Eintrag für 31. projektierten Adressbereich
Eintrag für 32. projektierten Adressbereich

Bild 6-4 Aufbau der kennungsbezogenen Diagnose des IM 151-7 CPU

6.6.5 Modulstatus

Definition

Der Modulstatus gibt den Status der projektierten Adressbereiche wieder und stellt eine Detaillierung der kennungsbezogenen Diagnose bezüglich der Konfiguration dar. Der Modulstatus beginnt nach der kennungsbezogenen Diagnose und umfasst maximal 13 Byte.

Aufbau

Der Modulstatus des IM 151-7 CPU ist wie folgt aufgebaut:

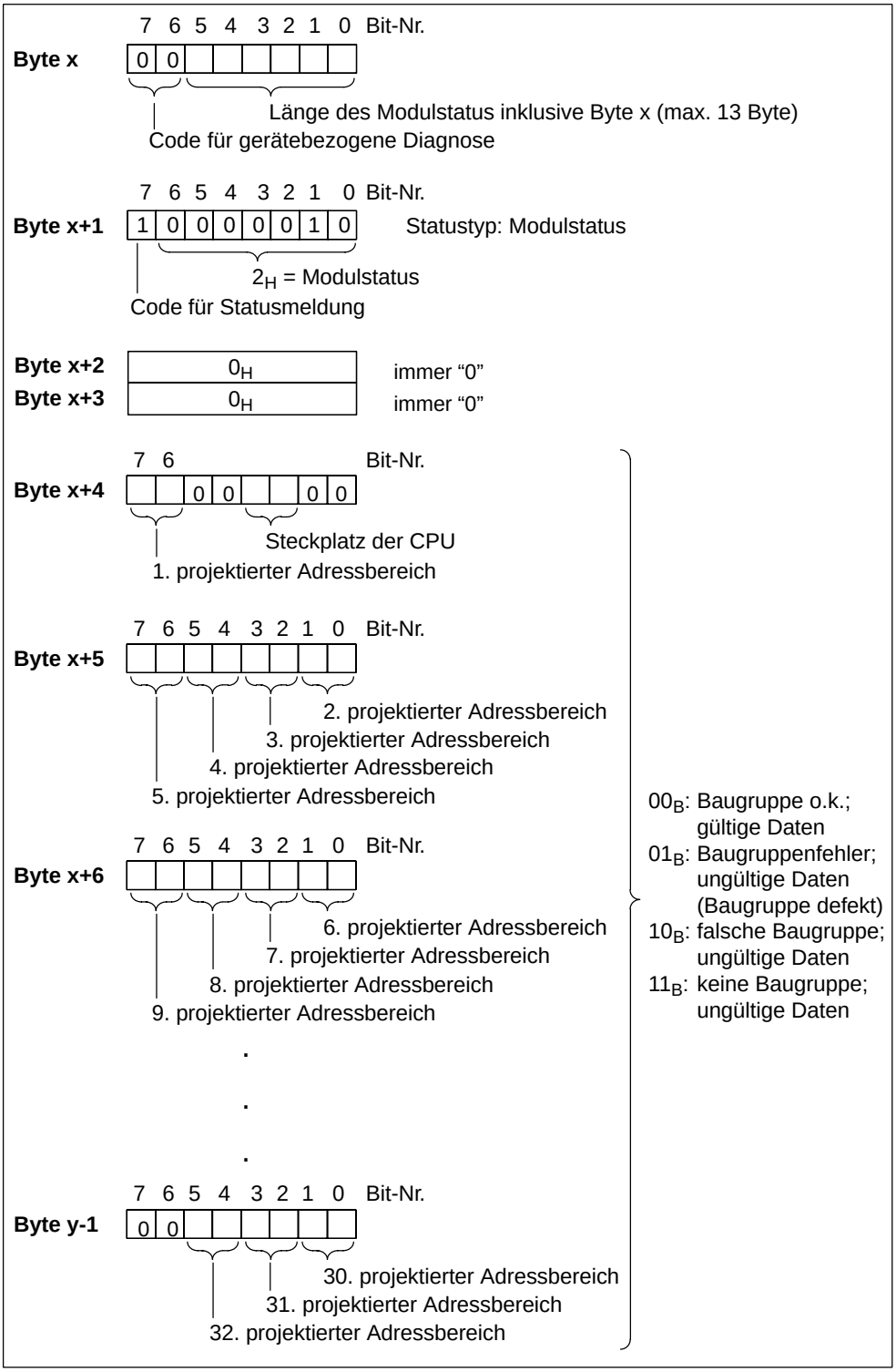


Bild 6-5 Aufbau des Modulstatus

6.6.6 Alarmstatus

Definition

Der Alarmstatus der gerätebezogenen Diagnose gibt detaillierte Auskunft über einen DP-Slave. Die gerätebezogene Diagnose beginnt ab Byte y und kann maximal 20 Byte umfassen.

Aufbau

Im folgenden Bild sind Aufbau und Inhalt der Bytes für einen projektierten Adressbereich des Übergabespeichers beschrieben.

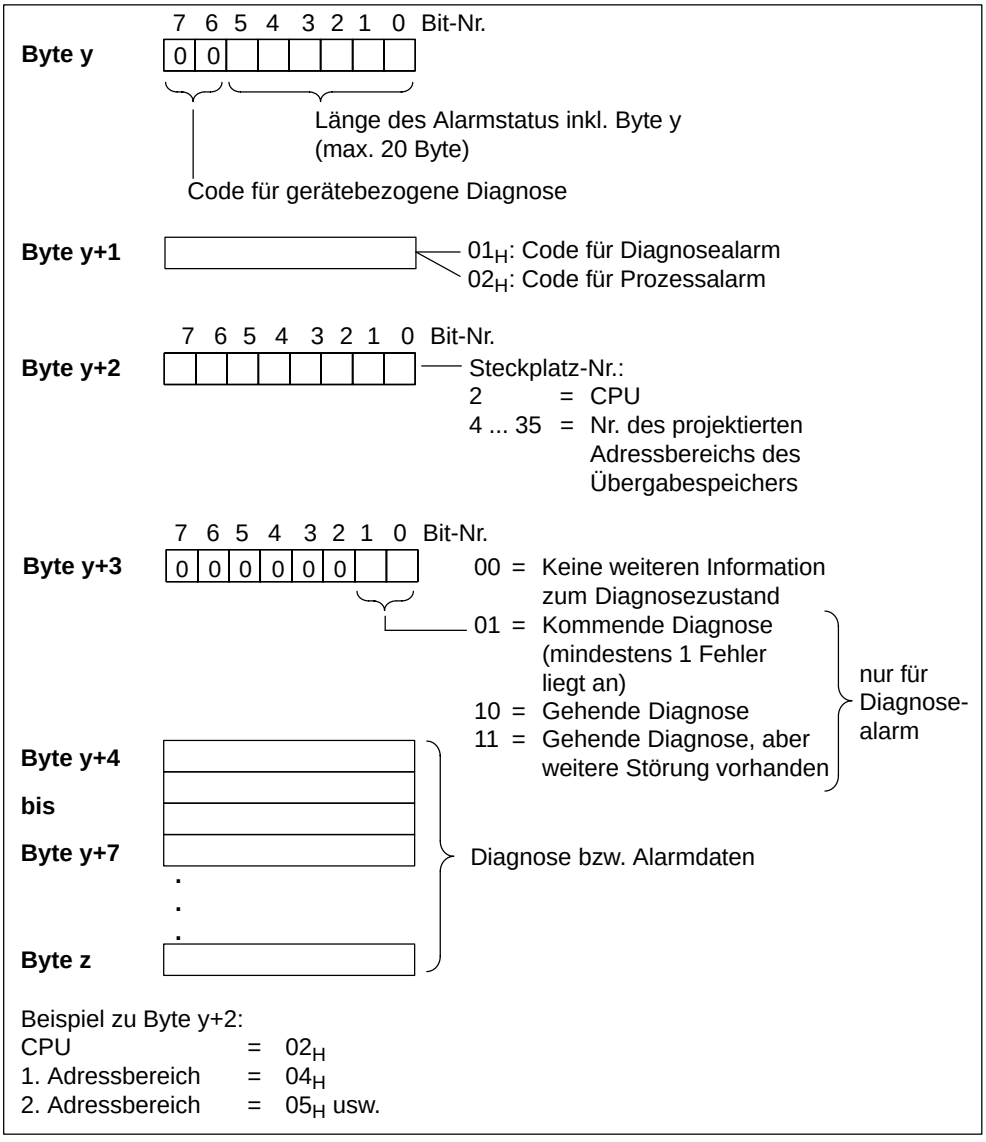

7 6 5 4 3 2 1 0 Bit-Nr.
Byte y
0 0
Länge des Alarmstatus inkl. Byte y
(max. 20 Byte)
Code für gerätebezogene Diagnose
Byte y+1
01H: Code für Diagnosealarm
02H: Code für Prozessalarm
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+2
Steckplatz-Nr.:
2 = CPU
4 ... 35 = Nr. des projektierten Adressbereichs des Übergabespeichers
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+3
0 0 0 0 0 0
00 = Keine weiteren Information zum Diagnosezustand
01 = Kommende Diagnose (mindestens 1 Fehler liegt an)
10 = Gehende Diagnose
11 = Gehende Diagnose, aber weitere Störung vorhanden
nur für Diagnose-alarm
Byte y+4
bis
Byte y+7
Diagnose bzw. Alarmdaten
Byte z
Beispiel zu Byte y+2:
CPU = 02H
1. Adressbereich = 04H
2. Adressbereich = 05H usw.

Bild 6-6 Aufbau des Alarmstatus

Aufbau der Alarmdaten bei Prozessalarm (ab Byte y+4)

Beim Prozessalarm (in Byte y+1 steht Code 02_H für Prozessalarm) werden ab Byte y+4 die 4 Byte Alarminformationen übergeben, die Sie im I-Slave mit dem SFC 7 “DP_PRAL” bzw. SFB 75 “SALRM” beim Generieren des Prozessalarms für den Master übergeben.

Aufbau der Alarmdaten bei Erzeugung eines Diagnosealarms durch einen Betriebszustandswechsel des I-Slave (ab Byte y+4)

Im Byte y+1 steht der Code für Diagnosealarm (01_H). Die Diagnosedaten enthalten die 16 Byte Zustandsinformation der CPU. Im folgenden Bild sehen Sie die Belegung der ersten 4 Byte der Diagnosedaten. Die folgenden 12 Byte sind immer 0.

Die Inhalte dieser Bytes entsprechen dem Inhalt des Datensatzes 0 der Diagnose in *STEP 7* (in diesem Fall sind nicht alle Bits belegt).

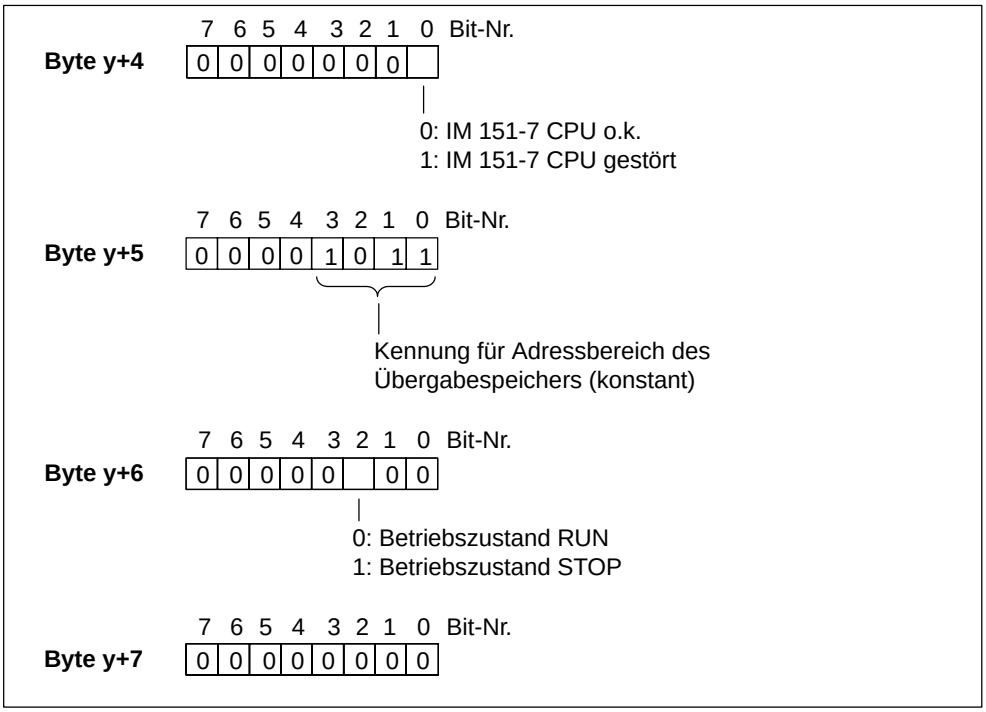

7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+4
0 0 0 0 0 0 0
0: IM 151-7 CPU o.k.
1: IM 151-7 CPU gestört
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+5
0 0 0 0 1 0 1 1
Kennung für Adressbereich des Übergabespeichers (konstant)
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+6
0 0 0 0 0 0 0
0: Betriebszustand RUN
1: Betriebszustand STOP
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+7
0 0 0 0 0 0 0 0

Bild 6-7 Byte y+4 bis y+7 für Diagnosealarm (Betriebszustandswechsel des I-Slave)

Aufbau der Alarmdaten bei Erzeugung eines Diagnosealarms durch den SFB 75 im I-Slave (ab Byte y+4)

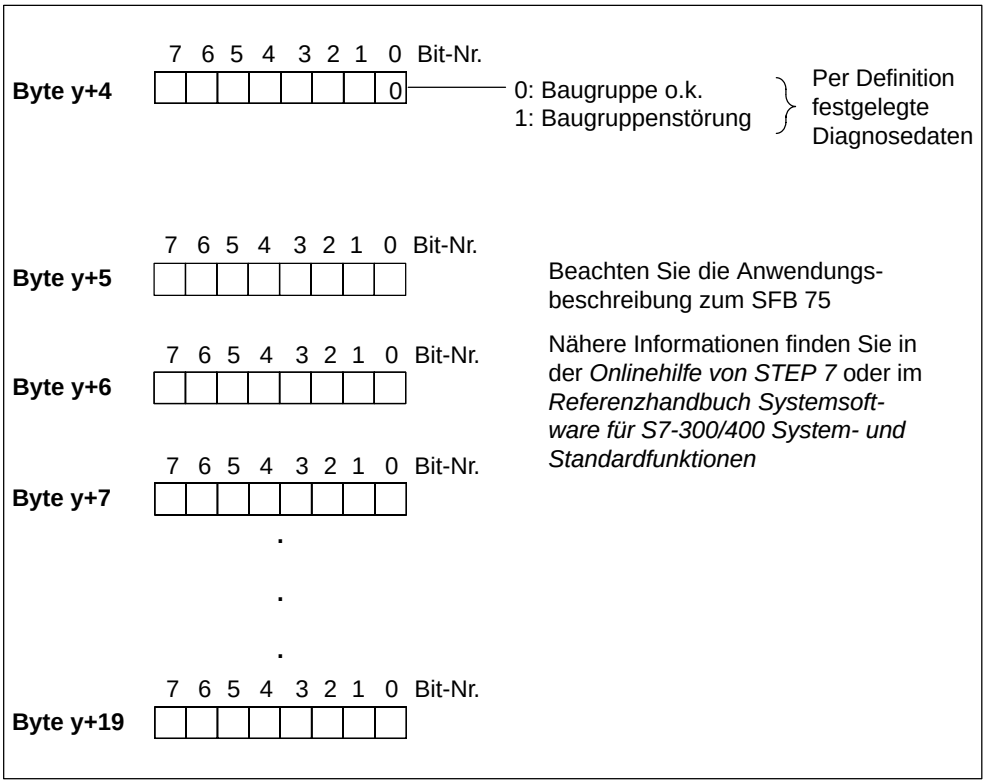

7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+4
0
0: Baugruppe o.k.
1: Baugruppenstörung
Per Definition festgelegte Diagnosedaten
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+5
Beachten Sie die Anwendungs-beschreibung zum SFB 75
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+6
Nähere Informationen finden Sie in der Onlinehilfe von STEP 7 oder im Referenzhandbuch Systemsoft-ware für S7-300/400 System- und Standardfunktionen
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+7
7 6 5 4 3 2 1 0 Bit-Nr.
Byte y+19

Bild 6-8 Byte y+4 bis y+7 für Diagnosealarm (SFB 75)

6.7 Diagnosedaten der Elektronikmodule

6.7.1 Diagnosedaten der Elektronikmodule im Anwenderprogramm auswerten

In diesem Kapitel ...

ist der Aufbau der Diagnosedaten in den Systemdaten beschrieben. Diesen Aufbau müssen Sie kennen, wenn Sie im *STEP 7*-Anwenderprogramm die Diagnosedaten der Elektronikmodule auswerten wollen.

Diagnosedaten stehen in Datensätzen

Die Diagnosedaten eines Moduls können bis zu 44 Byte lang sein und stehen in den Datensätzen 0 und 1:

- Der Datensatz 0 enthält 4 Byte Diagnosedaten, die den aktuellen Zustand eines Automatisierungssystems beschreiben.
  Der DS0 ist Bestandteil der Kopfinformation des OB 82 (Lokaldatenbytes 8 bis 11).
- Der Datensatz 1 enthält die 4 Byte Diagnosedaten, die auch im Datensatz 0 stehen **und** bis zu 40 Byte baugruppenspezifische Diagnosedaten.

Sie können DS0 und DS1 über die SFC 59 “RD_REC” bzw. SFB 52 “RDREC” auslesen.

Weiterführende Literatur

Eine umfassende Beschreibung des Prinzips der Auswertung der Diagnosedaten von Elektronikmodulen im Anwenderprogramm sowie die Beschreibung der dafür anwendbaren SFCs finden Sie in den Handbüchern zu *STEP 7*.

Aufbau der Diagnosedaten

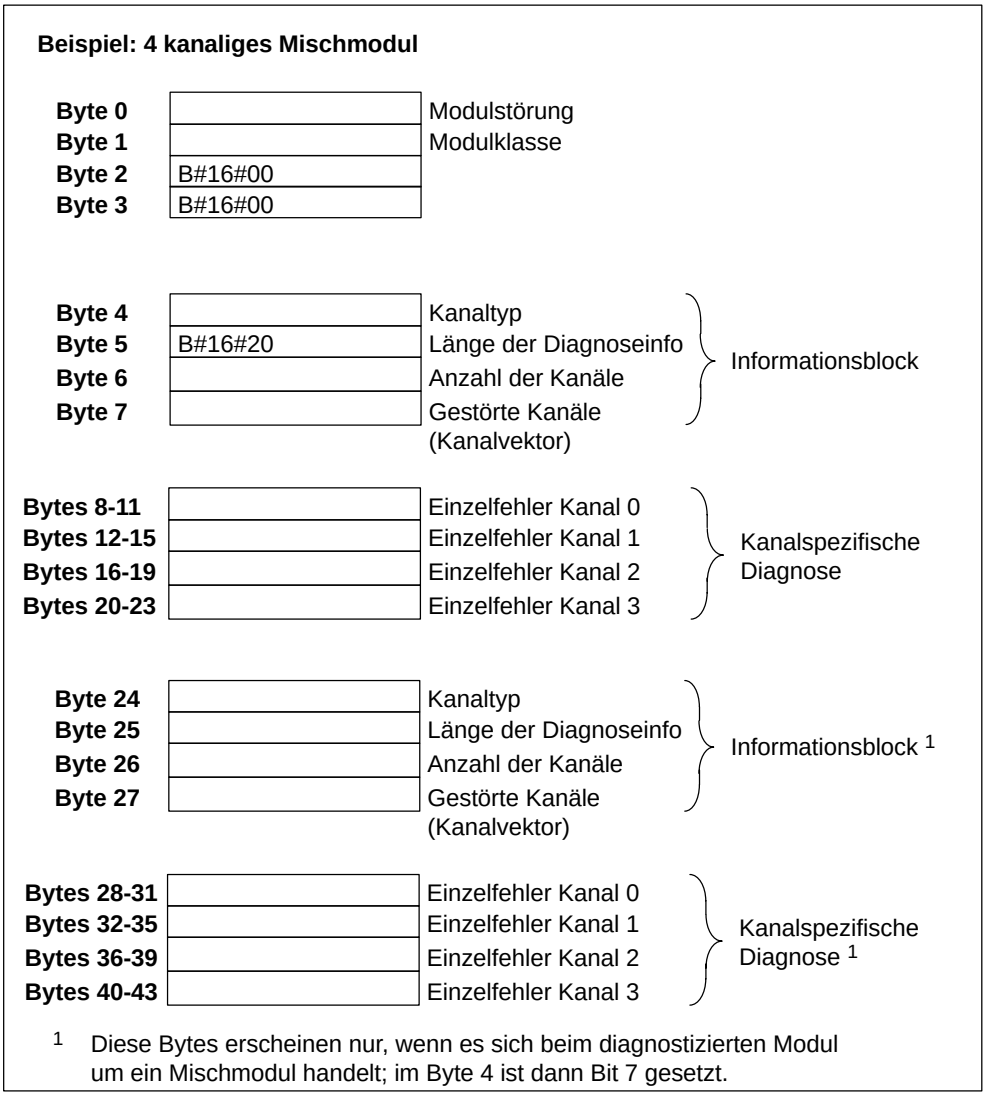


Bild 6-9 Aufbau der Diagnosedaten am Beispiel eines 4 kanaligen Mischmoduls

Die Anzahl der kanalspezifischen Diagnosebytes hängt von der Anzahl der Kanäle im Modul ab. Kanal 0 ist aber mindestens vorhanden. Deshalb beträgt die Minimallänge des DS1 12 Byte.

Wenn Sie z. B. ein Mischmodul mit 1 Eingangs- und 2 Ausgangskanälen haben, beginnt der zweite Informationsblock bei Byte 12. Die Gesamtlänge der Diagnosedaten beträgt in diesem Beispiel 24 Byte.

6.7.2 Aufbau und Inhalt der Diagnosedaten Bytes 0 bis 7

Nachfolgend sind Aufbau und Inhalt der einzelnen Bytes der Diagnosedaten beschrieben. Generell gilt: Wenn ein Fehler auftritt, dann wird das entsprechende Bit auf “1” gesetzt.

Bytes 0 und 1

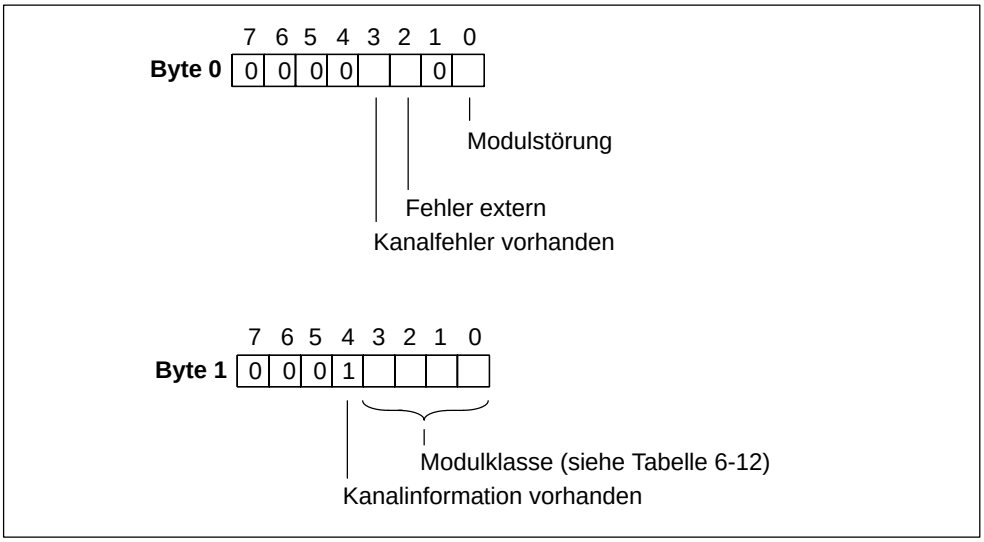


Bild 6-10 Bytes 0 und 1 der Diagnosedaten

Modulklassen

Die folgende Tabelle enthält die Kennungen der Modulklassen (Bits 0 bis 3 im Byte 1).

Tabelle 6-12 Kennungen der Modulklassen

Kennung	Modulklasse
0101	Analogmodul
0110	CPU
1000	Funktionsmodul
1001	Digitalmodul (Peripherie mit eingeschränktem Adressraum)
1100	CP
1101	PS

Bytes 2 und 3

Diese Bytes werden nicht genutzt.

Bytes 4 bis 7

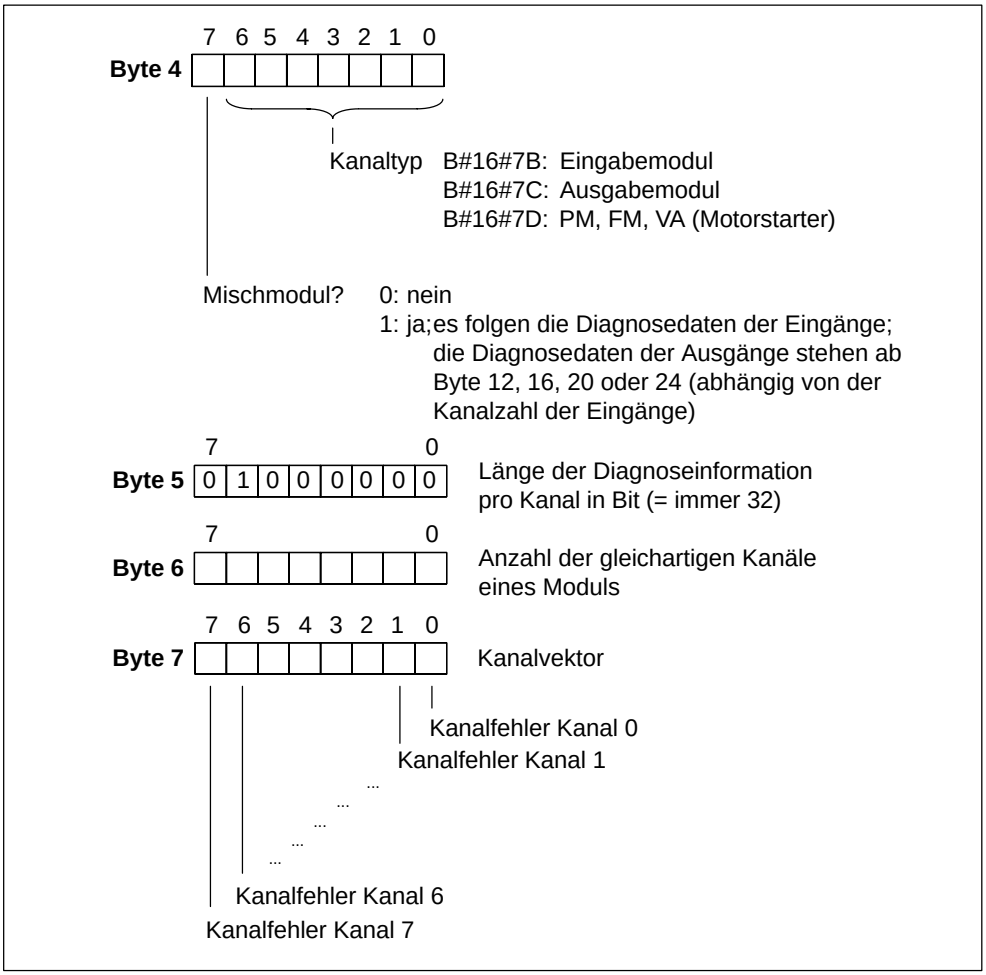


Bild 6-11 Bytes 4 bis 7 der Diagnosedaten

6.7.3 Kanalspezifische Diagnosedaten ab Byte 8

Ab Byte 8 enthält der Datensatz 1 die kanalspezifischen Diagnosedaten. Die folgenden Bilder zeigen die Belegung des Diagnosebytes für einen Kanal bzw. eine Kanalgruppe des speziellen Moduls. Generell gilt: Wenn ein Fehler auftritt, dann wird das entsprechende Bit auf “1” gesetzt.

Einzelfehler eines Kanales

“Byte y” ist das erste von vier Byte der kanalspezifischen Diagnose eines Kanals.

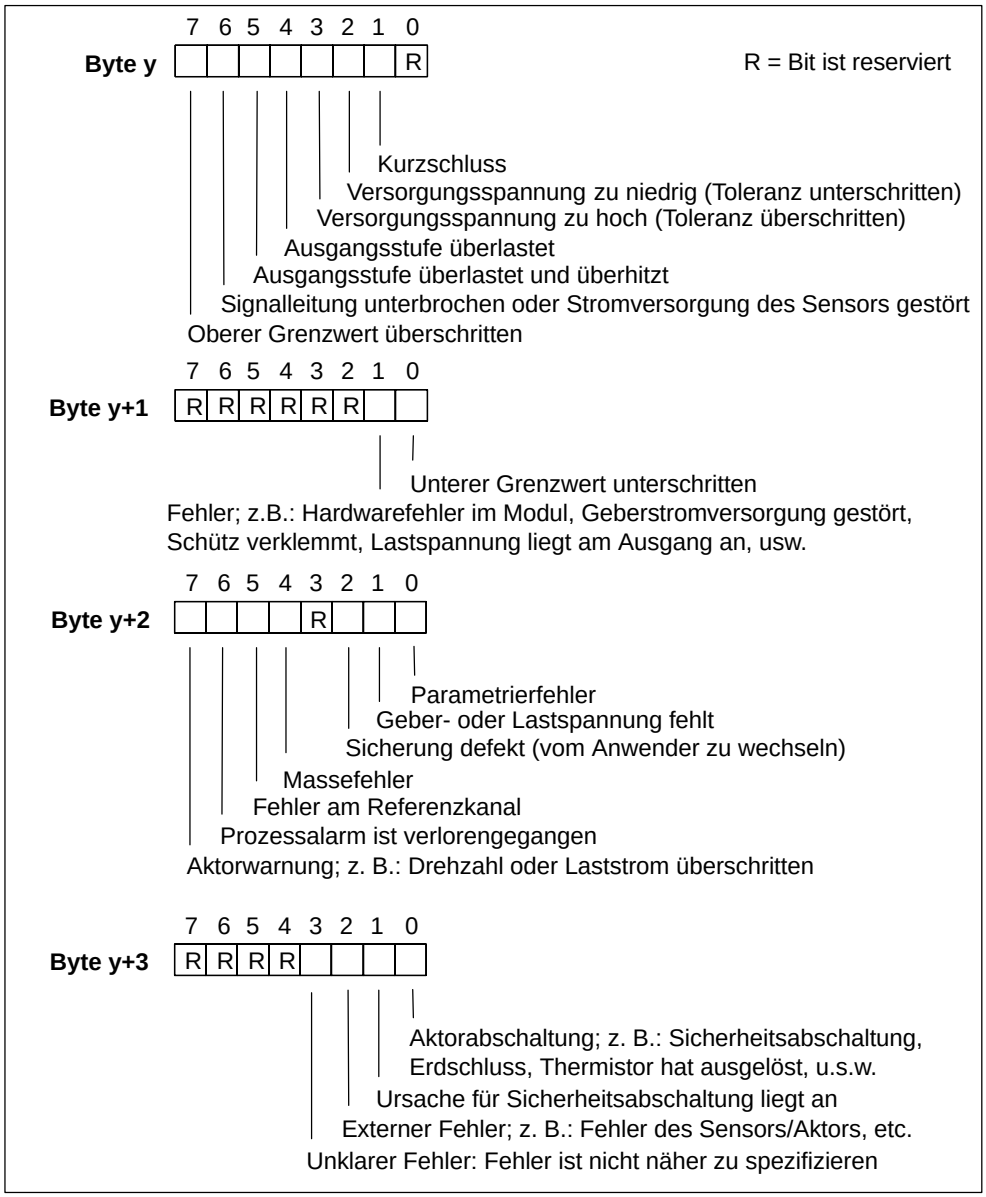

7 6 5 4 3 2 1 0
Byte y
R
R = Bit ist reserviert
Kurzschluss
Versorgungsspannung zu niedrig (Toleranz unterschritten)
Versorgungsspannung zu hoch (Toleranz überschritten)
Ausgangsstufe überlastet
Ausgangsstufe überlastet und überhitzt
Signalleitung unterbrochen oder Stromversorgung des Sensors gestört
Oberer Grenzwert überschritten
7 6 5 4 3 2 1 0
Byte y+1
R R R R R R
Unterer Grenzwert unterschritten
Fehler; z.B.: Hardwarefehler im Modul, Geberstromversorgung gestört,
Schütz verklemmt, Lastspannung liegt am Ausgang an, usw.
7 6 5 4 3 2 1 0
Byte y+2
R
Parametrierfehler
Geber- oder Lastspannung fehlt
Sicherung defekt (vom Anwender zu wechseln)
Massefehler
Fehler am Referenzkanal
Prozessalarm ist verlorengegangen
Aktorwarnung; z. B.: Drehzahl oder Laststrom überschritten
7 6 5 4 3 2 1 0
Byte y+3
R R R R
Aktorabschaltung; z. B.: Sicherheitsabschaltung,
Erdschluss, Thermistor hat ausgelöst, u.s.w.
Ursache für Sicherheitsabschaltung liegt an
Externer Fehler; z. B.: Fehler des Sensors/Aktors, etc.
Unklarer Fehler: Fehler ist nicht näher zu spezifizieren

Bild 6-12 Einzelfehler eines Kanales

6.7.4 Beispiel: ET 200S-Modul: 2 AI U (6ES7 134-4FB00-0AB0) mit je einer Diagnose für Kanal 0 und 1

Die folgende Tabelle enthält beispielhaft die Auswertung einer Diagnosemeldung des genannten Moduls.

Byte Nummer	Wert	Bedeutung
0	B#16#0D	Modulstörung, Fehler extern, Kanalfehler vorhanden
1	B#16#15	Kanalinformation vorhanden; Typklasse = Analogmodul
2	B#16#00	nicht genutzt
3	B#16#00	nicht genutzt
4	B#16#7B	Eingabemodul, kein Mischmodul
5	B#16#20	= 32 Bit Diagnoseinformation pro Kanal (konstant)
6	B#16#02	Das Modul besitzt 2 Kanäle
7	B#16#03	Kanalfehler an Kanal 0 und Kanal 1
8	B#16#80	Kanalfehler Kanal 0: Oberer Grenzwert ist überschritten
9	B#16#00	Kanal 0: kein weiterer Fehler
10	B#16#00	Kanal 0: kein weiterer Fehler
11	B#16#00	Kanal 0: kein weiterer Fehler
12	B#16#00	Kanal 1: kein Fehler
13	B#16#01	Kanalfehler Kanal 1: Unterer Grenzwert ist unterschritten
14	B#16#00	Kanal 1: kein weiterer Fehler
15	B#16#00	Kanal 1: kein weiterer Fehler

Funktionen des IM 151-7 CPU

7

In diesem Kapitel

Sie finden in diesem Kapitel:

- wichtige Eigenschaften des IM 151-7 CPU für PROFIBUS-DP
- eine Aufstellung von CPU-Funktionen des IM 151-7 CPU, die Sie mit *STEP 7* aufrufen können, wie die integrierte Uhr, Bausteine für das Anwenderprogramm und einstellbare Parameter

Kapitelübersicht

Kapitel	Thema	Seite
7.1	Daten für PROFIBUS-DP	7-2
7.2	Betriebsartenschalter und Anzeigeelemente	7-4
7.3	SIMATIC Micro Memory Card	7-6
7.4	Speicherkonzept	7-12
7.5	Schnittstellen	7-29
7.6	Uhr	7-31
7.7	Kommunikation	7-32
7.8	Bausteine	7-36
7.9	Parameter	7-39
7.10	Parametrierung der Vergleichsstelle beim Anschluss von Thermo-elementen	7-41
7.11	Ziehen und Stecken von Modulen im laufenden Betrieb	7-43
7.12	Aus- und Einschalten von Powermodulen im laufenden Betrieb	7-46

7.1 Daten für PROFIBUS-DP

GSD-Datei

In einer Geräte-Stammdaten-Datei (GSD-Datei) sind alle slavespezifischen Eigenschaften hinterlegt. Der Aufbau der GSD-Datei ist in der Norm IEC 61784-1:2002 Ed1 CP 3/1 festgelegt.

Sie benötigen die GSD-Datei nur dann, wenn Sie:

- ET 200S mit einem DP-Master aus dem SIMATIC S5-Spektrum einsetzen (Projektierung mit COM PROFIBUS)
- ET 200S mit einem SIMATIC-fremden DP-Master einsetzen (Projektierung mit Fremdtool)

Falls Sie die GSD-Datei benötigen, können Sie die GSD-Datei vom Internet herunterladen. Sie finden sämtliche GSD-Dateien unter “Downloads” auf der Internetseite des SIMATIC Customer Supports:

- http://www.ad.siemens.de/csi/gsd

Alternativ können Sie die GSD-Dateien über Modem unter der Telefonnummer +49 (0)911-737972 oder unter CompuServe im AUTFORUM (`GO AUTFORUM`) abrufen.

Wichtige Eigenschaften

Falls Sie die GSD-Datei nicht zur Hand haben, sind im Folgenden tabellarisch die wichtigsten Eigenschaften des IM 151-7 CPU aufgelistet.

Tabelle 7-1 Eigenschaften aus der GSD-Datei

Eigenschaft	DP-Schlüsselwort nach IEC 61784-1:2002 Ed1 CP 3/1	IM 151-7 CPU
Herstellerkennung	Ident_Number	$80E2_H$ $80.._H$ (FO)
Unterstützung von FMS	FMS_supp	nein
Unterstützung von 9,6 kBaud	9.6_supp	ja
Unterstützung von 19,2 kBaud	19.2_supp	ja
Unterstützung von 45,45 kBaud	45.45_supp	ja
Unterstützung von 93,75 kBaud	93.75_supp	ja
Unterstützung von 187,5 kBaud	187.5_supp	ja
Unterstützung von 500 kBaud	500_supp	ja
Unterstützung von 1,5 MBaud	1.5M_supp	ja
Unterstützung von 3 MBaud	3M_supp	ja nein (FO)

Tabelle 7-1 Eigenschaften aus der GSD-Datei

Eigenschaft	DP-Schlüsselwort nach IEC 61784-1:2002 Ed1 CP 3/1	IM 151-7 CPU
Unterstützung von 6 MBaud	6M_supp	ja nein (FO)
Unterstützung von 12 MBaud	12M_supp	ja
Unterstützung des Steuerkommandos FREEZE	Freeze_Mode_supp	ja
Unterstützung des Steuerkommandos SYNC	Sync_Mode_supp	ja
Unterstützung von automatischer Baudratensuche	Auto_Baud_supp	ja
PROFIBUS-Adresse über Software änderbar	Set_Slave_Add_supp	nein
Länge der anwenderspezifischen Parametrierdaten	User_Prm_Data_Len	3 Byte
Anwenderspezifische Parametrierdaten	User_Prm_Data	ja
Mindestabstand zwischen zwei Slave-Listen-umläufen	Min_Slave_Intervall	1 (100 µs)
Modulares Gerät	Modular_Station	1
Maximale Anzahl der Module	Max_Module	35
Maximale Zahl der Eingänge in Bytes	Max_Input_Len	244
Maximale Zahl der Ausgänge in Bytes	Max_Output_Len	244
Maximale Zahl der Ein- und Ausgänge zusammen in Bytes	Max_Data_Len	488
Zentrale Anzeige von herstellerspezifischen Status- und Fehlermeldungen	Unit_Diag_Bit	über LED “ON”
Zuordnung von Werten im gerätebezogenen Dia-gnosefeld zu Texten	Unit_Diag_Area	nicht genutzt
Kennungen aller Adressbereiche für PROFIBUS	Module, End_Module	ja
Zuordnung von herstellerspezifischen Fehlertypen im kanalbezogenen Diagnosefeld zu Texten	Channel_Diag	nein
Maximale Länge der Diagnosedaten	Max_Diag_Data_Len	39 Byte

7.2 Betriebsartenschalter und Anzeigeelemente

Betriebsartenschalter

Der Betriebsartenschalter des IM 151-7 CPU ist als 3-stufiger Kippschalter realisiert und hat folgendes Aussehen:

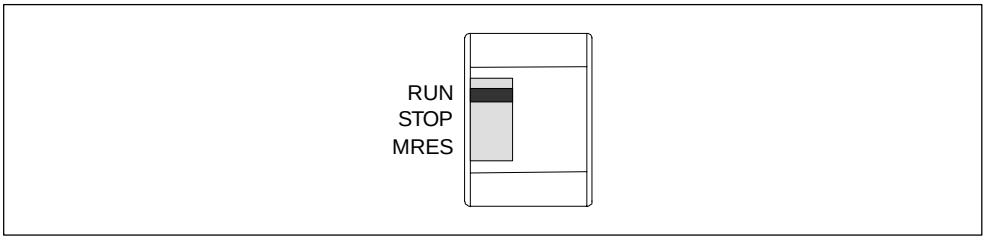


Bild 7-1 Betriebsartenschalter

Stellungen des Betriebsartenschalters

Die Stellungen des Betriebsartenschalters sind in der Reihenfolge erläutert, wie sie auf dem IM 151-7 CPU angeordnet sind.

Tabelle 7-2 Stellungen des Betriebsartenschalters

Stellung	Bedeutung	Erläuterungen
RUN	Betriebsart RUN	Der CPU-Teil bearbeitet das Anwenderprogramm.
STOP	Betriebsart STOP	Der CPU-Teil bearbeitet kein Anwenderprogramm. Programme können • mit PG aus dem CPU-Teil ausgelesen werden (CPU → PG) • in den CPU-Teil übertragen werden (PG → CPU)
MRES	Urlöschen	Taststellung des Betriebsartenschalters für das Urlöschen des CPU-Teils. Das Urlöschen per Betriebsartenschalter erfordert eine spezielle Bedienungsreihenfolge (siehe Kapitel 6.2)

Bedeutung der LEDs für CPU-Funktionalität

Für den CPU-Teil des IM 151-7 CPU gibt es 2 spezielle LEDs, die die Betriebszustände des CPU-Teils anzeigen:

- RUN
- STOP

Über 3 weitere LEDs können Sie Informationen zur Spannungsversorgung der CPU, zu Force-Aufträgen und zu allgemeinen Fehlern erhalten.

Tabelle 7-3 LED-Anzeige für CPU-Funktionalität

Anzeige	Bedeutung	Erläuterungen
ON (grün)	Netz-Ein	**leuchtet**, wenn die CPU-Versorgungsspannung anliegt
RUN (grün)	Betriebszustand RUN	**leuchtet**, wenn der CPU-Teil das Anwenderprogramm bearbeitet **blinkt mit 2 Hz** während des Anlaufs des CPU-Teils • für mindestens 3 s; der Anlauf des CPU-Teils kann aber kürzer sein • während des Anlaufs des CPU-Teils leuchtet zusätzlich die STOP-Anzeige; nach dem Erlöschen der STOP-Anzeige sind die Ausgänge freigegeben **blinkt mit 0,5 Hz**, wenn die CPU einen von Ihnen gesetzten Haltepunkt erreicht hat. Gleichzeitig leuchtet die STOP-LED
STOP (gelb)	Betriebszustand STOP	**leuchtet**, wenn der CPU-Teil • kein Anwenderprogramm bearbeitet • einen von Ihnen gesetzten Haltepunkt erreicht hat. Gleichzeitig blinkt die RUN-LED mit 0,5 Hz. **blinkt** mit 0,5 Hz, wenn der CPU-Teil “Urlöschen” anfordert (siehe Kapitel 6.2)
FRCE (gelb)	Force-Auftrag aktiv	**leuchtet**, wenn Force-Auftrag aktiv ist
SF (rot)	Sammelfehler	**leuchtet** bei • Programmierfehlern • Parametrierfehlern • Rechenfehlern • Zeitfehlern • Peripheriefehlern • Hardwarefehlern • Firmwarefehlern Zur genauen Fehlerermittlung müssen Sie ein PG einsetzen und den Diagnosepuffer auslesen.

Bedeutung weiterer LEDs

Die LEDs “SF” (aus PROFIBUS-DP-Sicht) und “BF” sind im Kapitel 6.4 beschrieben.

7.3 SIMATIC Micro Memory Card

Micro Memory Card

Als Speichermodul für das IM 151-7 CPU wird eine SIMATIC Micro Memory Card (MMC) verwendet. Die MMC ist als Ladespeicher und transportabler Datenträger einsetzbar. Sie ist zum Betrieb des IM 151-7 CPU zwingend erforderlich. Folgende Daten werden auf der MMC abgelegt:

- Anwenderprogramm (alle Bausteine)
- Archive und Rezepturen
- Projektierungsdaten (*STEP 7*-Projekte)
- Daten für ein Betriebssystem-Update, Sicherung des Betriebssystems

Hinweis

Auf **einer** MMC können Sie **entweder** Anwender- und Projektierungsdaten oder das Betriebssystem speichern.

Eigenschaften

Die SIMATIC Micro Memory Card stellt die Wartungsfreiheit und Remanenz für das IM 151-7 CPU sicher. Ausführlichere Informationen dazu finden Sie im Kapitel 7.4.

Vorsicht

Der Modulinhalt einer SIMATIC Micro Memory Card kann ungültig werden, wenn sie während eines laufenden Schreibvorganges entfernt wird. Die MMC muss dann ggf. am PG gelöscht bzw. im IM 151-7 CPU formatiert werden.

Entfernen Sie die MMC nie im Betriebszustand RUN, sondern nur im Netz-Aus oder im Zustand STOP des IM 151-7 CPU, wenn keine schreibenden PG-Zugriffe stattfinden. Wenn Sie im STOP nicht sicherstellen können, dass keine schreibenden PG-Funktionen (z. B. Baustein laden/löschen) aktiv sind, trennen Sie vorher die Kommunikationsverbindungen.

Lebensdauer einer MMC

Die Lebensdauer einer MMC hängt wesentlich von folgenden Faktoren ab:

1. Der Anzahl der Lösch- bzw. Programmiervorgänge,
2. äußeren Einflüssen wie beispielsweise der Umgebungstemperatur.

Bei einer Umgebungstemperatur von bis zu 60° C beträgt die Lebensdauer einer MMC bei maximal 100.000 Lösch-/Schreibvorgängen 10 Jahre.

Vorsicht

Achten Sie immer darauf, die maximale Anzahl der Lösch-/Schreibvorgänge nicht zu überschreiten, um Datenverlusten vorzubeugen.

Einsetzbare SIMATIC Micro Memory Cards

Es stehen Ihnen folgende Speichermodule zur Verfügung:

Tabelle 7-4 Verfügbare MMCs

Typ	Bestellnummern
MMC 64k	6ES7 953-8LF00-0AA0
MMC 128k	6ES7 953-8LG00-0AA0
MMC 512k	6ES7 953-8LJ00-0AA0
MMC 2M	6ES7 953-8LL00-0AA0
MMC 4M	6ES7 953-8LM00-0AA0
MMC 8M	6ES7 953-8LP10-0AA0

Für ein Firmware-Update sind die MMCs mit 4 MByte bzw. 8 MByte Speicher notwendig.

Formatierung der MMC beim Urlöschen

In bestimmten Sonderfällen müssen Sie die MMC formatieren:

- Der Modultyp ist kein Anwendermodul.
- Die MMC wurde noch nicht formatiert, ist defekt oder die Daten sind inkonsistent.

Der Inhalt der MMC wurde als ungültig gekennzeichnet.

- Der Vorgang *Anwenderprogramm laden* wurde durch Netz-Aus abgebrochen (siehe Sonderhandling).
- Der Vorgang *Prommen* wurde durch Netz-Aus abgebrochen (siehe Sonderhandling).
- Fehler bei der Auswertung des Modulinhaltes beim Urlöschen.
- Fehler bei der Formatierung, bzw. Formatierung konnte nicht ausgeführt werden.

Wenn einer dieser beschriebenen Fehler aufgetreten ist, fordert die CPU auch nach Ausführen eines Urlöschvorganges wieder erneut Urlöschen an. Ausser bei Unterbrechung der Vorgänge Anwenderprogramm laden oder Prommen durch Netz-Aus bleibt der Karteninhalt bis zur Ausführung des Sonderhandlings erhalten.

Beschreibung Sonderhandling:

Wenn das IM 151-7 CPU in Urlöschanforderung steht (langsames Blinken der STOP-LED), formatieren Sie diese mit folgender Schalterbedienung:

1. Bringen Sie den Schalter in Stellung MRES und halten Sie ihn so lange fest (ca. 9 Sekunden), bis die STOP-LED dauerhaft leuchtet.
2. Innerhalb der nächsten 3 Sekunden müssen Sie den Schalter loslassen und erneut in die Stellung MRES bringen. Die STOP-LED blinkt nun während der Formatierung.

Achten Sie darauf, die Schritte in der vorgeschriebenen Zeit durchzuführen, da die MMC sonst nicht formatiert wird, sondern in den Zustand Urlöschen zurückfällt.

Die MMC wird nur formatiert, wenn ein Formatierungsgrund (s. o.) vorliegt; nicht z. B. bei Urlöschanforderung nach Modultausch. Hier bewirkt ein Schalten auf MRES nur ein normales Urlöschen, bei dem der Modulinhalt gültig bleibt.

Stecken/Wechseln der Karte

Die MMC ist so ausgelegt, dass sie auch unter Spannung gezogen und gesteckt werden kann. Dabei muss das IM 151-7 CPU in den STOP-Zustand geschaltet werden (siehe Warnhinweis auf Seite 7-6). Durch die abgeschrägte Ecke der MMC wird verhindert, dass die Karte verkehrt herum gesteckt werden kann (Verpolschutz).

Um Ihnen ein Herausnehmen der Karte zu ermöglichen, ist die Fassung des Modulschachtes mit einem Auswerfer versehen. Zum Auswerfen der Karte drücken Sie mit einem kleinen Schraubendreher oder Kugelschreiber auf den Auswerfer.

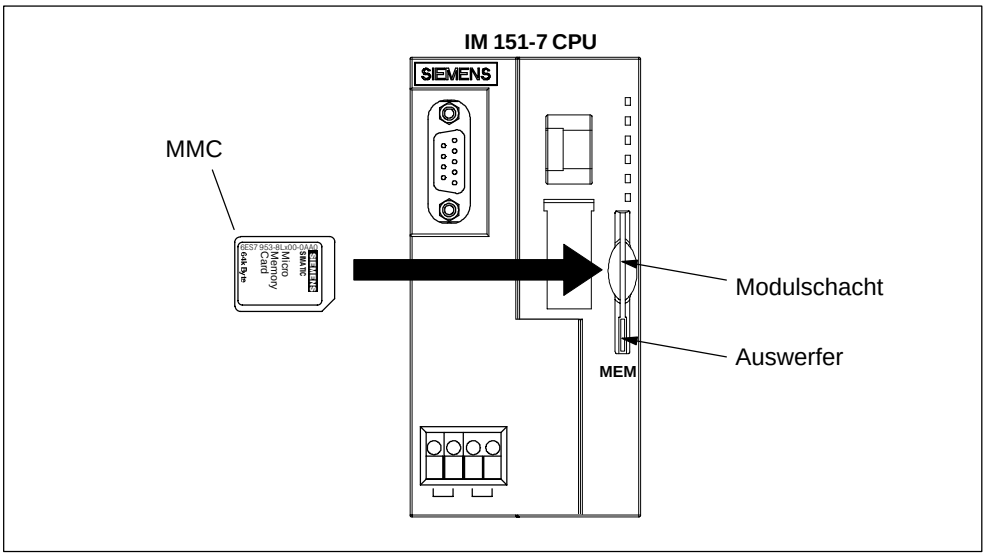


Bild 7-2 Position des Modulschachtes für die MMC-Karte am IM 151-7 CPU

Wird eine neue MMC in den Modulschacht gesteckt, fordert das IM 151-7 CPU Urlöschen an.

Firmware-Update mit MMC

Ein Update der Firmware führen Sie folgendermaßen durch:

Tabelle 7-5 Firmware-Update mit MMC

Schritt	Das müssen Sie tun:	Das passiert im IM 151-7 CPU:
1.	Update-Dateien mittels STEP 7 und Ihrem Programmiergerät auf eine leere MMC ($\geq$ 4 MB) übertragen.	-
2.	IM 151-7 CPU spannungsfrei schalten und MMC mit FW-Update stecken.	-
3.	Spannung einschalten.	IM 151-7 CPU erkennt die MMC mit dem FW-Update automatisch und startet das FW-Update. Während des FW-Update leuchten alle LEDs. Nach Abschluss des FW-Update blinkt die STOP-LED. Das IM 151-7 CPU fordert damit Urlöschen an.
4.	IM 151-7 CPU spannungsfrei schalten und MMC mit FW-Update ziehen.	-
5.	Schalten Sie die Spannungsversorgung wieder ein.	IM 151-7 CPU führt ein automatisches Urlöschen durch und ist danach betriebsbereit.

Sichern des Betriebssystems auf MMC

Das Sichern des Betriebssystems erfordert folgende Vorgehensweise:

Tabelle 7-6 Sichern der Betriebssystems

Schritt	Das müssen Sie tun:	Das passiert im IM 151-7 CPU:
1.	Neue Micro Memory Card (≥ 4 MB) in die CPU stecken .	CPU fordert Urlöschen an.
2.	Betriebsartenschalter in der Stellung MRES halten.	–
3.	NETZ-AUS/NETZ-EIN und Betriebsartenschalter in Stellung MRES halten bis ...	... STOP-, RUN- und FRCE-LEDs zu blinken beginnen.
4.	Betriebsartenschalter auf STOP.	–
5.	Betriebsartenschalter kurzzeitig nach MRES bewegen, dann wieder nach STOP schnappen lassen.	Das IM 151-7 CPU beginnt, das Betriebssystem auf MMC zu sichern. Während der Sicherung leuchten alle LEDs. Nach Abschluss der Sicherung blinkt die STOP-LED. Das IM 151-7 CPU fordert damit Urlöschen an.
6.	Micro Memory Card ziehen.	–

7.4 Speicherkonzept

7.4.1 Speicherbereiche des IM 151-7 CPU

Aufteilung

Der Speicher des IM 151-7 CPU lässt sich in drei Bereiche aufteilen:

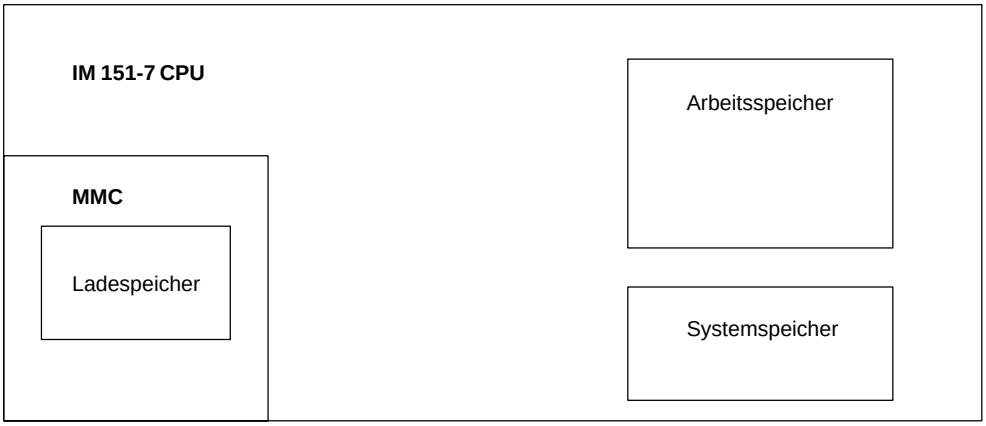


Bild 7-3 Speicherbereiche eines IM 151-7 CPU

Ladespeicher

Der Ladespeicher ist auf einer SIMATIC Micro Memory Card (MMC) untergebracht. Er dient zur Aufnahme von Code- und Datenbausteinen sowie von Systemdaten (Konfiguration, Verbindungen, Baugruppenparameter, usw.).

Bausteine, die als nicht ablaufrelevant gekennzeichnet sind, werden ausschließlich in den Ladespeicher aufgenommen.

Zusätzlich können die kompletten Projektierungsdaten eines Projekts auf der MMC abgelegt werden.

Ihr Programm im Ladespeicher (MMC) ist immer remanent. Es wird bereits beim Laden netzausfallsicher und urlöschfest auf der MMC hinterlegt.

Hinweis

Der Betrieb des IM 151-7 CPU ist **nur mit gesteckter MMC** möglich.

Arbeitsspeicher

Der Arbeitsspeicher ist auf der CPU integriert und ist nicht erweiterbar. Er dient zur Abarbeitung des Codes sowie zur Bearbeitung der Daten des Anwenderprogramms. Die Programmbearbeitung erfolgt ausschließlich im Bereich von Arbeitsspeicher und Systemspeicher.

Mit gesteckter MMC ist der Arbeitsspeicher der CPU remanent.

Ihre Daten im Arbeitsspeicher werden bei Netz-Aus auf der MMC gesichert.

Systemspeicher

Der Systemspeicher ist auf der CPU integriert und ist nicht erweiterbar.

Er enthält

- die Operandenbereiche Merker, Zeiten und Zähler
- die Prozessabbilder der Ein- und Ausgänge
- die Lokaldaten

Für Merker, Zeiten und Zähler bestimmen Sie durch Projektierung (Eigenschaften der CPU, Register Remanenz), welche Teile remanent sein sollen und welche bei Neustart (Warmstart) mit “0” initialisiert werden sollen.

Diagnosepuffer, MPI-Adresse (und Baudrate) sowie Betriebsstundenzähler sind generell im remanenten Speicherbereich auf der CPU abgelegt. Mit der Remanenz der MPI-Adresse und Baudrate wird sichergestellt, dass Ihre CPU nach einem Stromausfall, nach Urlöschen oder nach Verlust der Kommunikationsparametrierung (durch Ziehen der MMC oder Löschen der Kommunikationsparameter) noch kommunikationsfähig ist.

Remanenz

Das IM 151-7 CPU besitzt remanenten Speicher. Die Remanenz wird auf der MMC und auf der CPU realisiert.

Durch die Remanenz bleibt der Inhalt von remanentem Speicher auch über NETZ-AUS und Neustart (Warmstart) hinweg erhalten.

Remanenzverhalten der Speicherobjekte

Nachfolgende Tabelle zeigt das Remanenzverhalten der Speicherobjekte bei den einzelnen Betriebszustandsübergängen.

Tabelle 7-7 Remanenzverhalten der Speicherobjekte

Speicherobjekt	Betriebszustandsübergang		
	NETZ-EIN/ NETZ-AUS	**STOP → RUN**	**Urlöschen**
Anwenderprogramm/-daten (Ladespeicher)	X	X	X
Aktualwerte der DBs	X	X	–
als remanent projektierte Merker, Zeiten und Zähler	X	X	–
Diagnosepuffer, Betriebsstundenzähler	X	X	X
MPI-Adresse, Baudrate	X	X	X

x = remanent; – = nicht remanent

7.4.2 Speicherfunktionen

Einleitung

Mit Hilfe der Speicherfunktionen erzeugen, modifizieren oder löschen Sie Anwenderprogramme bzw. einzelne Bausteine. Nutzen Sie weiterhin die Möglichkeit, eigene Projektdaten zu archivieren und sorgen Sie auf diesem Weg für die Remanenz Ihrer Daten.

Allgemein: Anwenderprogramm per PG/PC laden

Das Anwenderprogramm wird komplett **per PG/PC über die MMC auf das IM 151-7 CPU** geladen. Dabei werden unter Umständen alle im Ladespeicher vorhandenen Bausteine gelöscht.

Bausteine belegen im Ladespeicher den Platz, wie er unter “Ladespeicherbedarf” in den “Allgemeinen Bausteineigenschaften” genannt wird.

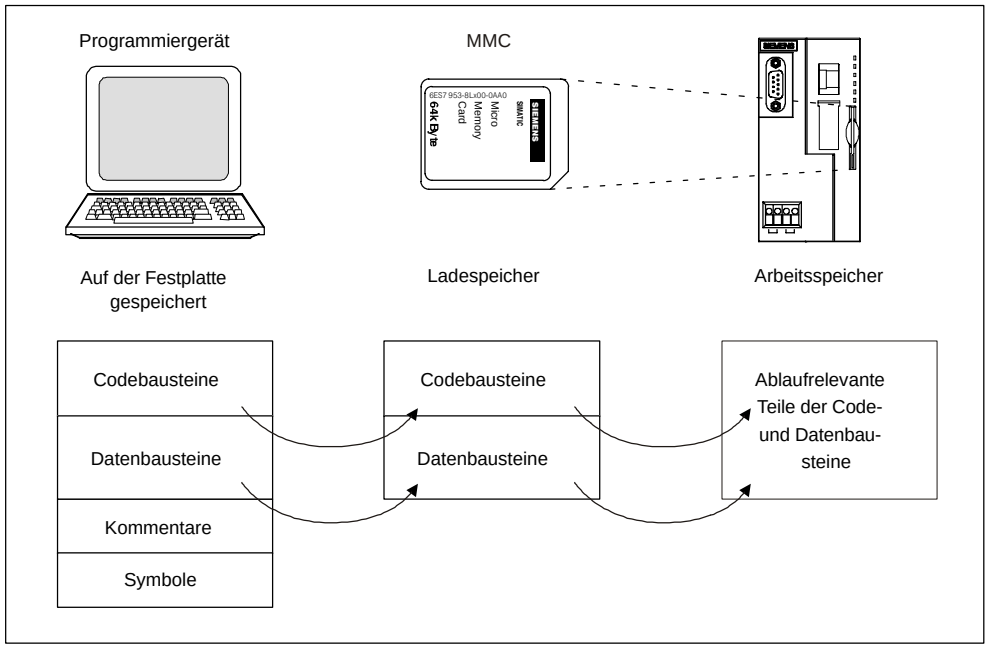


Bild 7-4 Lade- und Arbeitsspeicher

Erst nachdem alle Bausteine geladen sind, kann das Programm gestartet werden.

Hinweis

Die Funktion ist nur im STOP der CPU zulässig.

Wenn der Ladevorgang durch Netzausfall oder unzulässige Bausteine nicht beendet werden konnte, ist anschließend der Ladespeicher leer.

Ein Anwenderprogramm per PG/PC auf die MMC laden

Fall A: Laden eines neuen Anwenderprogramms
Sie haben ein neues Anwenderprogramm erstellt. Dieses laden Sie vollständig per PG/PC auf die MMC.

Fall B: Nachladen von Bausteinen
Sie haben bereits ein Anwenderprogramm erstellt und auf die MMC geladen (Fall A). Im folgenden erweitern Sie das Anwenderprogramms um weitere Bausteine. Dazu müssen Sie das Anwenderprogramm nicht erneut vollständig auf die MMC laden, sondern nur die neuen Bausteine auf die MMC nachladen (bei sehr komplexen Programmen verkürzt sich bei dieser Vorgehensweise die Ladezeit!).

Fall C: Überladen
In diesem Fall nehmen Sie Änderungen an Bausteinen Ihres Anwenderprogramms vor. Im nächsten Schritt überladen Sie dann das Anwenderprogramm bzw. nur veränderte Bausteine per PG/PC auf die MMC.

Warnung

Beim Überladen von Bausteinen/eines Anwenderprogramms gehen alle auf der MMC unter gleichem Namen gespeicherten Daten verloren.

Nach Laden eines Bausteins wird bei ablaufrelevanten Bausteinen der Inhalt in den Arbeitsspeicher übertragen und aktiviert.

Löschen von Bausteinen

Beim Löschen wird der Baustein aus dem Ladespeicher gelöscht. Das Löschen von Datenbausteinen ist aus dem Anwenderprogramm möglich (SFC 23 “DEL_DB”).

Ist durch diesen Baustein Speicher im Arbeitsspeicher belegt worden, wird dieser freigegeben.

Hochladen

Im Gegensatz zum Vorgang Laden wird unter dem Hochladen das Laden einzelner Bausteine oder eines vollständigen Anwenderprogramms **von der CPU in das PG/PC** verstanden. Die Bausteine haben dabei den Inhalt des letzten Ladens in die MMC. Ausnahme bilden ablaufrelevante Datenbausteine, bei ihnen werden die Aktualwerte übertragen.

Das Hochladen von Bausteinen oder des Anwenderprogramms aus der CPU mit *STEP 7* hat keine Auswirkung auf die Speicherbelegung der CPU.

Komprimieren

Beim Komprimieren werden durch Lade- und Löschvorgänge im Lade- und Arbeitsspeicher entstandene Lücken zwischen Speicherobjekten geschlossen. Damit wird der freie Speicher zusammenhängend zur Verfügung gestellt.

Komprimieren ist sowohl im STOP als auch im RUN der CPU möglich.

Prommen (RAM to ROM)

Beim Prommen werden aus dem Arbeitsspeicher die Aktualwerte der Datenbausteine als neue Anfangswerte der DB in den Ladespeicher übernommen.

Hinweis

Die Funktion ist nur im STOP der CPU zulässig.

Wenn die Funktion durch Netzausfall nicht beendet werden konnte, ist anschließend der Ladespeicher leer.

Ziehen/Stecken der MMC

Ist keine MMC auf dem IM 151-7 CPU gesteckt, so ist das IM 151-7 CPU nicht lauffähig (kein Ladespeicher vorhanden). Erst nach Stecken einer MMC und Urlöschen ist ein sinnvoller Betrieb möglich.

Ziehen und Stecken einer MMC erkennt das IM 151-7 CPU in jedem Betriebszustand.

Ablauf beim Ziehen:

1. Das IM 151-7 CPU muss in STOP geschaltet werden.
2. Es dürfen keine schreibenden PG-Funktionen aktiv sein (z. B. Laden von Bausteinen)
3. Nach Ziehen der MMC fordert das IM 151-7 CPU Urlöschen an

Vorsicht

Der Modulinhalt einer SIMATIC Micro Memory Card kann ungültig werden, wenn sie während eines laufenden Schreibvorganges entfernt wird. Die MMC muss dann ggf. am PG gelöscht bzw. im IM 151-7 CPU formatiert werden.

Entfernen Sie die MMC nie im Betriebszustand RUN, sondern nur im Netz-Aus oder im Zustand STOP des IM 151-7 CPU, wenn keine schreibenden PG-Zugriffe stattfinden. Wenn Sie im STOP nicht sicherstellen können, dass keine schreibenden PG-Funktionen (z. B. Baustein laden/löschen) aktiv sind, trennen Sie vorher die Kommunikationsverbindungen.

Ablauf beim Stecken:

Das Stecken der MMC mit dem zugehörigen Anwenderprogramm läuft in folgenden Schritten ab:

1. MMC stecken
2. IM 151-7 CPU fordert Urlöschen an
3. Urlöschen quittieren

   Sollte das IM 151-7 CPU infolge einer falschen MMC oder einer MMC mit Firmware-Update erneut Urlöschen anfordern, dann finden Sie das weitere Vorgehen im Kapitel 7.3 unter *Sonderhandling* beschrieben.
4. IM 151-7 CPU starten

Warnung

Achten Sie darauf, dass die zu steckende MMC das zum IM 151-7 CPU (zur Anlage) passende Anwenderprogramm enthält. Ein falsches Anwenderprogramm kann zu schwerwiegenden Prozesswirkungen führen.

Urlöschen

Urlöschen stellt nach Ziehen/Stecken der Micro Memory Card wieder definierte Verhältnisse her, um einen Neustart (Warmstart) des IM 151-7 CPU zu ermöglichen.

Beim Urlöschen wird die Speicherverwaltung des IM 151-7 CPU neu aufgebaut. Alle Bausteine des Ladespeichers bleiben erhalten. Alle ablaufrelevanten Bausteine werden aus dem Ladespeicher erneut in den Arbeitsspeicher übernommen, insbesondere werden dadurch die Datenbausteine im Arbeitsspeicher initialisiert (erhalten wieder ihre Ladewerte aus dem Ladespeicher).

Der Vorgang des Urlöschens sowie Besonderheiten dazu sind im Kapitel 6.2 beschrieben.

Neustart (Warmstart)

- Alle DB behalten ihre Aktualwerte.
- Alle remanenten M, Z, T behalten ihre Werte.
- Alle nicht remanenten Anwenderdaten werden initialisiert:
  - M, Z, T, E, A mit “0”
- Alle Ablaufebenen setzen von vorne auf.
- Die Prozessabbilder werden gelöscht.

7.4.3 Operandenbereiche

Übersicht

Der Systemspeicher des IM 151-7 CPU ist in Operandenbereiche aufgeteilt (siehe nachfolgende Tabelle). Durch Verwendung der entsprechenden Operationen adressieren Sie in Ihrem Programm die Daten direkt in den jeweiligen Operandenbereich.

Tabelle 7-8 Operandenbereiche des Systemspeichers

Operandenbereiche	Beschreibung
Prozessabbild der Eingänge	Zu Beginn jedes OB 1-Zyklus liest das IM 151-7 CPU die Eingänge aus den Eingabebaugruppen und speichert die Werte in das Prozessabbild der Eingänge.
Prozessabbild der Ausgänge	Das Programm berechnet während des Zyklus die Werte für die Ausgänge und legt sie im Prozessabbild der Ausgänge ab. Am Ende des OB 1-Zyklus schreibt das IM 151-7 CPU die errechneten Ausgangswerte in die Ausgabebaugruppen.
Merker	Dieser Bereich stellt Speicherplatz für im Programm errechnete Zwischenergebnisse zur Verfügung.
Zeiten	In diesem Bereich stehen Zeiten zur Verfügung.
Zähler	In diesem Bereich stehen Zähler zur Verfügung.
Lokaldaten	Dieser Speicherbereich nimmt die temporären Daten eines Code-Bausteins (OB, FB, FC) für die Dauer der Bearbeitung dieses Bausteins auf.
Datenbausteine	Siehe Kapitel 7.4.4

Welche Adressbereiche bei Ihrer CPU möglich sind, entnehmen Sie der *Operationsliste* im Anhang A.

Prozessabbild der Ein- und Ausgänge

Werden im Anwenderprogramm die Operandenbereiche Eingänge (E) und Ausgänge (A) angesprochen, werden nicht die Signalzustände auf den digitalen Elektronikmodulen abgefragt, sondern es wird auf einen Speicherbereich im Systemspeicher der CPU zugegriffen. Diesen Speicherbereich bezeichnet man als Prozessabbild.

Das Prozessabbild ist in zwei Teile gegliedert: das Prozessabbild der Eingänge und das Prozessabbild der Ausgänge.

Vorteile des Prozessabbildes

Der Zugriff auf das Prozessabbild hat gegenüber dem direkten Zugriff auf die Elektronikmodule den Vorteil, dass der CPU für die Dauer der zyklischen Programmbearbeitung ein konsistentes Abbild der Prozesssignale zur Verfügung steht. Wenn sich während der Programmbearbeitung ein Signalzustand auf einem Eingabemodul ändert, bleibt der Signalzustand im Prozessabbild erhalten bis zur Prozessabbildaktualisierung im nächsten Zyklus. Außerdem benötigt der Zugriff auf das Prozessabbild wesentlich weniger Zeit als der direkte Zugriff auf die Elektronikmodule, weil sich das Prozessabbild im Systemspeicher der CPU befindet.

Aktualisieren des Prozessabbildes

Das Prozessabbild wird vom Betriebssystem zyklisch aktualisiert. Nachfolgendes Bild zeigt die Bearbeitungsschritte innerhalb eines Zyklus.

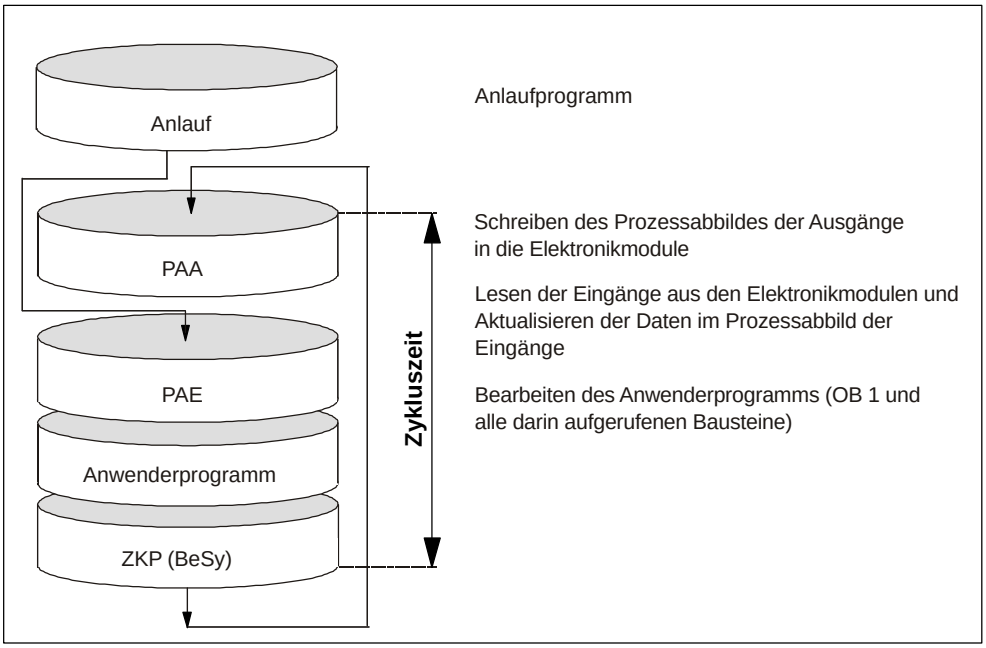

Anlauf
PAA
PAE
Anwenderprogramm
ZKP (BeSy)
Zykluszeit
Anlaufprogramm
Schreiben des Prozessabbildes der Ausgänge in die Elektronikmodule
Lesen der Eingänge aus den Elektronikmodulen und Aktualisieren der Daten im Prozessabbild der Eingänge
Bearbeiten des Anwenderprogramms (OB 1 und alle darin aufgerufenen Bausteine)

Bild 7-5 Bearbeitungsschritte innerhalb eines Zyklus

Lokaldaten

Die Lokaldaten speichern:

- die temporären Variablen von Code-Bausteinen
- die Startinformation der Organisationsbausteine
- Übergabeparameter
- Zwischenergebnisse

Temporäre Variablen

Beim Erstellen von Bausteinen können Sie temporäre Variablen (TEMP) deklarieren, die nur während der Bearbeitung des Bausteins zur Verfügung stehen und dann wieder überschrieben werden. Diese Lokaldaten haben pro OB eine feste Länge. Vor dem ersten lesenden Zugriff müssen die Lokaldaten initialisiert werden. Außerdem benötigt jeder Organisationsbaustein für seine Startinformation 20 Byte Lokaldaten. Der Zugriff auf Lokaldaten erfolgt schneller als auf Daten in DBs.

Das IM 151-7 CPU besitzt Speicher für die temporären Variablen (Lokaldaten) gerade bearbeiteter Bausteine. Er wird zu gleichen Teilen unter den Prioritätsklassen aufgeteilt. Jede Prioritätsklasse verfügt über einen eigenen Lokaldatenbereich.

Vorsicht

Alle temporären Variablen (TEMP) eines OB und seiner unterlagerten Bausteine werden in den Lokaldaten gespeichert. Wenn Sie viele Schachtelungsebenen in Ihrer Bausteinbearbeitung verwenden, kann der Lokaldatenbereich überlaufen. Das IM 151-7 CPU wechselt in den Betriebszustand STOP, wenn Sie die zulässige Größe der Lokaldaten einer Prioritätsklasse überschreiten.
Berücksichtigen Sie dabei den Lokaldatenbedarf von Synchronfehler-OBs, er wird jeweils der verursachenden Prioritätsklasse zugeordnet.

7.4.4 Handling von Daten in DB

Rezepturen

Eine Rezeptur ist eine Sammlung von Anwenderdaten.

Ein einfaches Rezepturkonzept lässt sich über nicht ablaufrelevante Datenbausteine realisieren. Dafür sollten die Rezepturen die gleiche Struktur (Länge) haben. Für jede Rezeptur sollte es einen DB geben.

Bearbeitungsablauf

Rezeptur soll im Ladespeicher abgelegt werden:

- Die einzelnen Datensätze der Rezepturen werden mit *STEP 7* als nicht ablaufrelevante DB erstellt und auf das IM 151-7 CPU geladen. Die Rezepturen belegen damit Platz nur im Ladespeicher und nicht im Arbeitsspeicher.

Arbeiten mit den Rezepturdaten:

- Mit der SFC 83 “READ_DBL” wird aus dem Anwenderprogramm heraus der Datensatz der aktuellen Rezeptur aus dem DB im Ladespeicher in einen ablaufrelevanten DB in den Arbeitsspeicher gelesen. Damit wird erreicht, dass der Arbeitsspeicher nur die Datenmenge eines Datensatzes aufnehmen muss.

  Jetzt kann das Anwenderprogramm auf die Daten der aktuellen Rezeptur zugreifen.

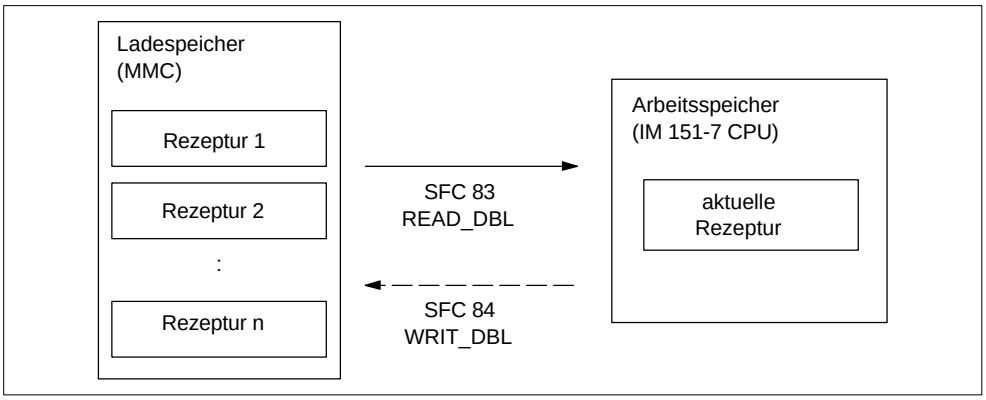


Bild 7-6 Handling von Rezepturdaten

Zurückspeichern einer geänderten Rezeptur:

- Mit der SFC 84 “WRIT_DBL” können aus dem Anwenderprogramm heraus neue bzw. ein geänderter Datensatz einer Rezeptur, die während der Programmbearbeitung entstanden ist, in den Ladespeicher zurückgeschrieben werden.

  Diese in den Ladespeicher geschriebenen Daten sind urlöschfest und transportabel.

  Sollen geänderte Datensätze (Rezepturen) auf dem PG/PC gesichert werden, so können sie als ganzer Baustein hochgeladen und dort gesichert werden.

Hinweis

Aktive Systemfunktionen SFC 82 bis 84 (laufende Zugriffe auf die MMC) haben starken Einfluss auf PG-Funktionen (z. B. Status Baustein, Status Variable, Baustein laden, hochladen, öffnen). Die Performance ist dabei (gegenüber nicht aktiven Systemfunktionen) typisch um den Faktor 10 niedriger.

Achten Sie immer darauf, die maximale Anzahl der Lösch-/Schreibvorgänge nicht zu überschreiten, um Datenverlusten vorzubeugen. Lesen Sie dazu auch das Kapitel 7.3.

Vorsicht

Der Modulinhalt einer SIMATIC Micro Memory Card kann ungültig werden, wenn sie während eines laufenden Schreibvorganges entfernt wird. Die MMC muss dann ggf. am PG gelöscht bzw. in der CPU formatiert werden.

Entfernen Sie die MMC nie im Betriebszustand RUN, sondern nur im Netz-Aus oder im Zustand STOP der CPU, wenn keine schreibenden PG-Zugriffe stattfinden. Wenn Sie im STOP nicht sicherstellen können, dass keine schreibenden PG-Funktionen (z. B. Baustein laden/löschen) aktiv sind, trennen Sie vorher die Kommunikationsverbindungen.

Messwertarchive

Bei der Bearbeitung des Anwenderprogramms durch das IM 151-7 CPU entstehen Messwerte. Diese Messwerte sollen archiviert und ausgewertet werden.

Bearbeitungsablauf

Sammeln der Messwerte:

- In einem DB (für Wechselpufferbetrieb in mehreren DB) werden vom IM 151-7 CPU die Messwerte im Arbeitsspeicher gesammelt.

Archivieren der Messwerte:

- Mit der SFC 84 “WRIT_DBL” können aus dem Anwenderprogramm heraus die Messwerte in DB in den Ladespeicher ausgelagert werden, bevor das Datenvolumen die Speicherkapazität des Arbeitsspeichers übersteigen würde.

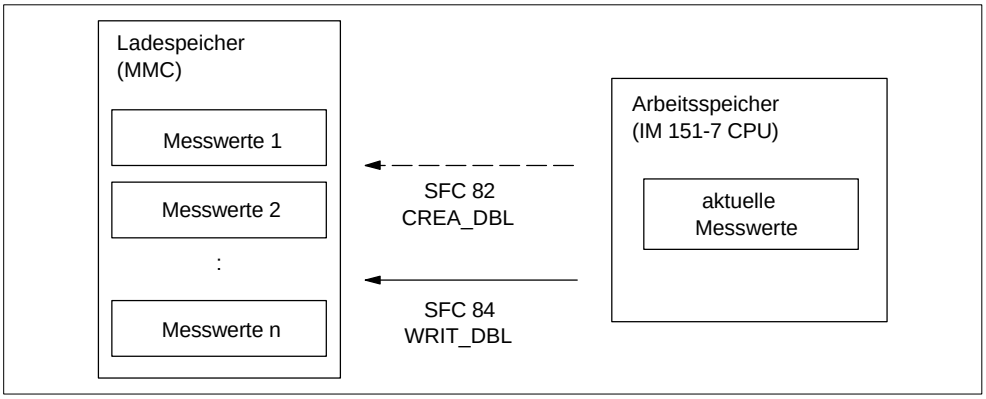


Bild 7-7 Handling von Messwertarchiven

- Mit der SFC 82 “CREA_DBL” können neue (zusätzliche) DB aus dem Anwenderprogramm heraus im Ladespeicher als nicht ablaufrelevante DB erzeugt werden, die keinen Platz im Arbeitsspeicher benötigen.

Hinweis

Ist bereits ein DB mit gleicher Nummer im Ladespeicher und/oder Arbeitsspeicher vorhanden, wird der SFC 82 beendet und eine Fehleranzeige generiert.

Diese in den Ladespeicher geschriebenen Daten sind urlöschfest und transportabel.

Auswerten der Messwerte:

- Die im Ladespeicher abgelegten Messwert-Datenbausteine können per Hochladen von anderen Kommunikationspartnern (z. B. PG, PC, ...) ausgewertet werden.

Hinweis

Aktive Systemfunktionen SFC 82 bis 84 (laufende Zugriffe auf die MMC) haben starken Einfluss auf PG-Funktionen (z. B. Status Baustein, Status Variable, Baustein laden, hochladen, öffnen). Die Performance ist dabei (gegenüber nicht aktiven Systemfunktionen) typisch um den Faktor 10 niedriger.

Achten Sie immer darauf, die maximale Anzahl der Lösch-/Schreibvorgänge nicht zu überschreiten, um Datenverlusten vorzubeugen. Lesen Sie dazu auch das Kapitel 7.3.

Vorsicht

Der Modulinhalt einer SIMATIC Micro Memory Card kann ungültig werden, wenn sie während eines laufenden Schreibvorganges entfernt wird. Die MMC muss dann ggf. am PG gelöscht bzw. in der CPU formatiert werden.

Entfernen Sie die MMC nie im Betriebszustand RUN, sondern nur im Netz-Aus oder im Zustand STOP der CPU, wenn keine schreibenden PG-Zugriffe stattfinden. Wenn Sie im STOP nicht sicherstellen können, dass keine schreibenden PG-Funktionen (z. B. Baustein laden/löschen) aktiv sind, trennen Sie vorher die Kommunikationsverbindungen.

7.4.5 Speichern/Holen ganzer Projekte auf/von Micro Memory Card

Arbeitsweise der Funktionen

Mit den Funktionen **Projekt auf Memory Card speichern** und **Projekt aus Memory Card holen** können Sie die kompletten Daten eines Projekts (für eine spätere Verwendung) auf einer SIMATIC Micro Memory Card speichern und wieder aus dieser zurückholen. Die SIMATIC Micro Memory Card kann sich hierfür in einer CPU oder in der MMC-Programmiereinrichtung eines PG bzw. PC befinden.

Hinweis

In der Online-Hilfe zu *STEP 7* sind die IM 151-7 CPU noch nicht berücksichtigt. Die Funktionen **Projekt auf Memory Card speichern** und Projekt aus Memory Card holen sind deshalb dort nur für die CPUs 41x relevant.
Mit den IM 151-7 CPU können Sie beide Funktionen im hier beschriebenen Umfang nutzen.

Die Projektdaten werden vor dem Speichern auf der SIMATIC Micro Memory Card komprimiert und beim Holen wieder dekomprimiert. Die Größe der Micro Memory Card muss so gewählt werden, dass neben den Projektdaten auch die Anwenderdaten gespeichert werden können. Die Größe der zu speichernden Projektdaten entspricht der Archivdateigröße dieses Projektes. Ist die Speicherkapazität der Micro Memory Card nicht ausreichend, so wird mit einer Meldung darauf hingewiesen.

Ist das Ziel für **Projekt auf Memory Card speichern** eine SIMATIC Micro Memory Card in einer CPU, so kann aus technischen Gründen nur der komplette Inhalt eines *STEP 7*-Projektes übertragen werden.

Umgang mit den Funktionen

Der Umgang mit den Funktionen **Projekt auf Memory Card speichern** / **Projekt aus Memory Card holen** ist davon abhängig, wo sich die SIMATIC Micro Memory Card befindet:

- Steckt die Micro Memory Card im MMC-Schacht eines IM 151-7 CPU, so selektieren Sie im Projektfenster des SIMATIC Managers eine Projektebene, die dem IM 151-7 CPU eindeutig zugeordnet ist (z. B. IM 151-7 CPU oder Programm oder Quellen oder Bausteine). Wählen Sie den Menübefehl **Zielsystem > Projekt auf Memory Card speichern** bzw. **Zielsystem > Projekt aus Memory Card holen**. Nun werden die kompletten Projektdaten auf die Micro Memory Card geschrieben bzw. aus dieser geholt.
- Sind die Projektdaten auf dem momentan genutzten Programmiergerät (PG/PC) nicht vorhanden, so kann die Quell-CPU im Fenster “Erreichbare Teilnehmer” ausgewählt werden. Öffnen Sie das Fenster “Erreichbare Teilnehmer” über den Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** und selektieren die gewünschte Verbindung/CPU mit den Projektdaten auf Micro Memory Card. Wählen Sie nun den Menübefehl **Projekt aus Memory Card holen**.
- Befindet sich die Micro Memory Card in der MMC-Programmiereinrichtung eines PG bzw. PC, so öffnen Sie das “S7-Memory Card-Fenster” mit dem Menübefehl **Datei > S7-Memory Card > Öffnen**. Wählen Sie den Menübefehl **Zielsystem > Projekt auf Memory Card speichern** bzw. **Zielsystem > Projekt aus Memory Card holen**. Ein Dialogfenster öffnet sich, über das Sie das Quell-Projekt bzw. das Ziel-Projekt anwählen können.

Projektdaten können ein sehr hohes Datenvolumen erzeugen, was gerade im Zustand RUN beim Lesen und Schreiben auf das IM 151-7 CPU zu Wartezeiten von mehreren Minuten führen kann.

Beispiel für einen Anwendungsfall

Sind im Service- und Instandhaltungsbereich mehrere Mitarbeiter mit der Wartung des Automatisierungssystems SIMATIC beauftragt, so ist es schwierig, jedem Mitarbeiter die aktuellen Projektdaten schnell für einen Serviceeinsatz zur Verfügung zu stellen. Sind die Projektdaten einer CPU jedoch lokal in einer der zu wartenden CPUs verfügbar, kann jeder Mitarbeiter auf die aktuellen Projektdaten zurückgreifen und gegebenenfalls Änderungen ausführen, die allen anderen Mitarbeitern wieder aktuell zur Verfügung stehen.

7.5 Schnittstelle

Das IM 151-7 CPU verfügt über eine koexistente Schnittstelle, die im Folgenden beschrieben ist.

MPI-Schnittstelle

Das MPI (Multi Point Interface) ist die Schnittstelle des IM 151-7 CPU zu einem PG/OP bzw. für die Kommunikation in einem MPI-Subnetz. Das IM 151-7 CPU verfügt über eine **MPI-Schnittstelle**, die mit RS 485 arbeitet.

Die typische (voreingestellte) Baudrate ist 187,5 kBaud. Die IM 151-7 CPU unterstützt alle MPI-Baudraten.

Das IM 151-7 CPU verschickt an der MPI-Schnittstelle automatisch ihre eingestellten Busparameter (z. B. die Baudrate). Damit kann sich beispielsweise ein Programmiergerät mit den richtigen Parametern versorgen und automatisch an ein MPI-Subnetz anschließen.

Hinweis

Im laufenden Betrieb dürfen Sie an das MPI-Subnetz nur PGs anschließen. Weitere Teilnehmer (z. B. OP, TP, ...) sollten Sie im laufenden Betrieb nicht mit dem MPI-Subnetz verbinden, da sonst die übertragenen Daten durch Störimpulse verfälscht werden oder Globaldaten-Pakete verloren gehen können.

PROFIBUS-DP-Schnittstelle

Die PROFIBUS-DP-Schnittstelle dient hauptsächlich zum Anschluss von dezentraler Peripherie. Mit PROFIBUS-DP können Sie ausgedehnte Subnetze aufbauen. Am PROFIBUS sind Baudraten bis 12 MBaud möglich.

Das IM 151-7 CPU als aktive Station verschickt an der PROFIBUS-DP-Schnittstelle die eingestellten Busparameter (z. B. die Baudrate). Damit kann sich beispielsweise ein Programmiergerät mit den richtigen Parametern versorgen und automatisch an ein PROFIBUS-Subnetz anschließen. Das Verschicken der Busparameter ist in der Projektierung abschaltbar.

Das IM 151-7 CPU verfügt über eine **PROFIBUS-DP-Schnittstelle**. Diese ist aktiv oder passiv konfigurierbar.

Welche Geräte können Sie an welche Schnittstelle anschließen?

Tabelle 7-9 Anschließbare Geräte

MPI	PROFIBUS-DP
• PG/PC • OP/TP • S7-300/400 mit MPI-Schnittstelle • S7-200 (nur mit 19,2 kBaud)	• DP-Master • Aktoren/Sensoren • S7-300/400 mit PROFIBUS-DP-Schnittstelle • PG/PC • OP/TP

Weiterführende Informationen

Weiterführende Informationen zu den einzelnen Verbindungen finden Sie im Handbuch *Kommunikation mit SIMATIC*.

7.6 Uhr

Das IM 151-7 CPU besitzt eine integrierte Hardware-Uhr.

Uhr stellen, lesen und programmieren

Die Uhr stellen und lesen Sie mit dem PG (siehe Benutzerhandbuch *STEP 7*) oder Sie programmieren die Uhr im Anwenderprogramm mit SFCs (siehe Referenzhandbuch *System- und Standardfunktionen* und Anhang B).

Eigenschaften

Die folgende Tabelle enthält die Eigenschaften und Funktionen der Uhr.

Über die Parametrierung des CPU-Teils in *STEP 7* können Sie auch Funktionen wie Synchronisation und Korrekturfaktor einstellen, lesen Sie dazu in der Online-Hilfe von *STEP 7* nach.

Tabelle 7-10 Eigenschaften der Uhr

Eigenschaften	IM 151-7 CPU
Typ	Hardware-Uhr
Voreinstellung bei Auslieferung	DT#1994-01-01-00:00:00
Pufferung	über eingebauten Kondensator
Pufferungsdauer	typ. 6 Wochen (bei 40 °C Umgebungstemperatur)
Betriebsstundenzähler	1

Verhalten der Uhr im NETZ-AUS

Die Uhr des CPU-Teils läuft über NETZ-AUS weiter.

Nach Ablauf der Pufferungsdauer läuft die Uhr mit der Uhrzeit weiter, bei der das NETZ-AUS erfolgte.

7.7 Kommunikation

Kommunikationsdienste des IM 151-7 CPU

Die Wahl des Kommunikationsdienstes hat Einfluss

- auf die Funktionalität, die dem Anwender zur Verfügung gestellt wird
- darauf, ob eine S7-Verbindung benötigt wird
- auf den Zeitpunkt des Verbindungsaufbaus

Die Anwenderschnittstelle kann sehr unterschiedlich sein (SFC, SFB, ...) und ist auch von der eingesetzten Hardware (SIMATIC-CPU, PC, ...) abhängig.

Das IM 151-7 CPU stellt Ihnen folgende Kommunikationsdienste zur Verfügung:

Tabelle 7-11 Kommunikationsdienste des IM 151-7 CPU

Kommunikationsdienst	Funktionalität	Aufbau der S7-Verbindung ...	über MPI	über DP
PG-Kommunikation	Inbetriebnahme, Test, Diagnose	Vom PG in dem Moment, wenn der Dienst benutzt wird	x	x
OP-Kommunikation	Bedienen und Beobachten	Vom OP beim Einschalten	x	x
S7-Basiskommunikation	Datenaustausch	Erfolgt programmiert über Bausteine (Parameter am SFC)	x	–
S7-Kommunikation	Datenaustausch	IM 151-7 CPU nur als Server; Verbindungsaufbau durch den Kommunikationspartner	x	x
Globale Datenkommunikation	zyklischer Austausch von Daten (z. B. Merker)	Benötigt keine S7-Verbindung	x	–

In den Kapiteln 4 und 5 finden Sie Informationen zum Netzaufbau und zur Adressierung.

PG-Kommunikation

Mit der PG-Kommunikation wird der Datenaustausch zwischen Engineering Stationen (z. B. PG, PC) und kommunikationsfähigen SIMATIC-Baugruppen realisiert. Der Dienst ist über MPI- und PROFIBUS-Subnetze möglich. Der Übergang zwischen Subnetzen wird ebenfalls unterstützt.

PG-Kommunikation stellt Funktionen zur Verfügung, die zum Laden von Programmen und Konfigurationsdaten, Durchführen von Tests und Auswerten von Diagnoseinformationen notwendig sind. Diese Funktionen sind im Betriebssystem der SIMATIC S7-Baugruppen integriert.

Eine CPU kann gleichzeitig mehrere Online-Verbindungen zu einem oder auch verschiedenen PGs halten.

OP-Kommunikation

Mit der OP-Kommunikation wird der Datenaustausch zwischen Operator Stationen (z. B. OP, TP) und kommunikationsfähigen SIMATIC-Baugruppen realisiert. Der Dienst ist über MPI- und PROFIBUS-Subnetze möglich.

OP-Kommunikation stellt Funktionen zur Verfügung, die zum Bedienen und Beobachten notwendig sind. Diese Funktionen sind im Betriebssystem der SIMATIC S7-Baugruppen integriert.

Eine CPU kann gleichzeitig mehrere Verbindungen zu einem oder auch verschiedenen OPs halten.

S7-Basiskommunikation

Mit der S7-Basiskommunikation wird der Datenaustausch zwischen S7-CPUs und kommunikationsfähigen SIMATIC-Baugruppen innerhalb einer S7-Station realisiert (quittierter Datenaustausch). Der Datenaustausch erfolgt über nichtprojektierte S7-Verbindungen. Der Dienst ist über das MPI-Subnetz oder in der Station zu Funktionsbaugruppen (FM) möglich.

S7-Basiskommunikation stellt Funktionen zur Verfügung, die zum Datenaustausch notwendig sind. Diese Funktionen sind im Betriebssystem des IM 151-7 CPU integriert.

Der Anwender kann den Dienst über die Anwenderschnittstelle “Systemfunktion” (SFC) nutzen.

S7-Kommunikation

IM 151-7 CPU kann nur Server in der S7-Kommunikation sein. Der Verbindungsaufbau erfolgt dabei immer vom Kommunikationspartner. Der Dienst ist über MPI- und PROFIBUS-Subnetze möglich.

Die Dienste werden vom Betriebssystem ohne explizite Anwenderschnittstelle abgewickelt.

Globale Datenkommunikation

Mit der Globalen Datenkommunikation wird der zyklische Austausch von Globaldaten (z. B. E, A, M) zwischen SIMATIC S7-CPUs realisiert (unquittierter Datenaustausch). Die Daten werden von einer CPU gleichzeitig an alle CPUs im MPI-Subnetz gesendet. Die Funktion ist im Betriebssystem des IM 151-7 CPU integriert.

Sende- und Empfangsbedingungen

Für die Kommunikation über GD-Kreise sollten Sie folgende Bedingungen einhalten:

- Für den Sender eines GD-Pakets muss gelten:
  $\text{Untersetzungsfaktor}_{\text{Sender}} \times \text{Zykluszeit}_{\text{Sender}} \geq 60\ \text{ms}$
- Für den Empfänger eines GD-Pakets muss gelten:
  $\text{Untersetzungsfaktor}_{\text{Empfänger}} \times \text{Zykluszeit}_{\text{Empfänger}} < \text{Untersetzungsfaktor}_{\text{Sender}} \times \text{Zykluszeit}_{\text{Sender}}$

Wenn Sie diese Bedingungen nicht einhalten, kann es zum Verlust eines GD-Pakets kommen. Gründe dafür sind:

- die Leistungsfähigkeit der “kleinsten” CPU im GD-Kreis
- das Senden und Empfangen von Globaldaten erfolgt asynchron durch Sender und Empfänger

Wenn Sie in *STEP 7* einstellen: “Senden nach jedem CPU-Zyklus” und die CPU hat einen kurzen CPU-Zyklus (< 60 ms), dann kann das Betriebssystem ein noch nicht gesendetes GD-Paket der CPU überschreiben. Der Verlust von Globaldaten wird im Statusfeld eines GD-Kreises angezeigt, wenn Sie dieses mit *STEP 7* projektiert haben.

Untersetzungsfaktor

Der Untersetzungsfaktor gibt an, auf wie viele Zyklen die GD-Kommunikation aufgeteilt wird. Den Untersetzungsfaktor können Sie bei der Projektierung der Globalen Datenkommunikation in *STEP 7* einstellen. Wenn Sie beispielsweise einen Untersetzungsfaktor von 7 wählen, erfolgt die Globale Datenkommunikation nur alle 7 Zyklen. Dadurch wird die CPU entlastet.

GD-Ressourcen

Die nachfolgende Tabelle gibt an, welche GD-Ressourcen das IM 151-7 CPU besitzt.

Tabelle 7-12 GD-Ressourcen des IM 151-7 CPU

Parameter	IM 151-7 CPU
Anzahl GD-Kreise je CPU	Max. 4
Anzahl Sende-GD-Pakete je GD-Kreis	Max. 1
Anzahl Sende-GD-Pakete für alle GD-Kreise	Max. 4
Anzahl Empfangs-GD-Pakete je GD-Kreis	Max. 1
Anzahl Empfangs-GD-Pakete für alle GD-Kreise	Max. 4
Datenlänge je GD-Paket	Max. 22 Byte
Konsistenz	Max. 22 Byte
Untersetzungsfaktor (default)	1 bis 255 (8)

Ausführliche Informationen ...

- zu SFCs finden Sie in der *Operationsliste*, eine ausführliche Beschreibung in der *Online-Hilfe zu STEP 7* oder im Referenzhandbuch *System- und Standardfunktionen*.
- zur Kommunikation finden Sie im Handbuch *Kommunikation mit SIMATIC*.

7.8 Bausteine

Sie finden in diesem Kapitel eine Übersicht, welche Bausteine im IM 151-7 CPU ablauffähig sind.

Das Betriebssystem ist auf eine ereignisgesteuerte Anwenderprogrammbearbeitung ausgelegt. Die folgenden Tabellen zeigen, welche Organisationsbausteine (OBs) das Betriebssystem bei welchem Ereignis automatisch aufruft.

Weitere Informationen

Die detaillierte Beschreibung der Bausteine und der nachfolgend aufgelisteten Startereignisse der OBs finden Sie im Referenzhandbuch *System- und Standardfunktionen*. Einen Überblick über die gesamte *STEP 7*-Dokumentation finden Sie im Handbuch *Dezentrales Peripheriesystem ET 200S*.

Übersicht aller Bausteine

Tabelle 7-13 Übersicht: Bausteine

Baustein	Anzahl	Bereich	maximale Größe	Bemerkung
OB	14	–	16 kByte	Eine Auflistung der möglichen OBs finden Sie im Anschluss an diese Tabelle.
FC	512	0-511		–
FB	512	0-511		–
DB	511	1-511		0 ist reserviert
SFC	61	–	–	Eine Auflistung aller SFCs des CPU-Teils finden Sie im Anhang B.1.
SFB	11	–	–	Eine Auflistung aller SFBs des CPU-Teils finden Sie im Anhang B.2.

In jedem IM 151-7 CPU sind maximal 1024 Bausteine (Anzahl FBs + FCs + DBs) ladbar.

SFC 55 “WR_PARM”, SFC 56 “WR_DPARM”, SFC 57 “PARM_MOD”, SFC 58 “WR_REC”

Der Einsatz der SFCs 55-58 in Verbindung mit Ihrem IM 151-7 CPU ist wegen der statischen Baugruppenparameter nicht sinnvoll.

Wenn Sie die SFCs 55-58 trotzdem in Verbindung mit Ihrem IM 151-7 CPU einsetzen, kann es zu einem Fehlverhalten des IM 151-7 CPU kommen.

OBs für Zyklus und Anlauf

Tabelle 7-14 OBs für Zyklus und Anlauf

Zyklus und Anlauf	aufgerufener OB	mögliche Startereignisse
Zyklus	OB 1	• erster OB 1 nach Neustart (NETZ-EIN oder STOP-RUN-Übergang des IM 151-7 CPU) • Beendigung des vorherigen Programmzyklus
Anlauf (STOP-RUN-Übergang)	OB 100	• manuelle Neustart-Anforderung (STOP-RUN über Betriebsartenschalter oder MPI-Kommando) • automatische Neustart-Anforderung (nach NETZ-AUS – NETZ-EIN-Übergang)

OBs für Alarme

Tabelle 7-15 OBs fürAlarme

Alarme	aufgerufener OB	mögliche Startereignisse
Uhrzeitalarm	OB 10	• automatischer Start nach Stellen und Aktivieren des Uhrzeitalarms mit *STEP 7* • Aktivieren über SFC 30 nach Stellen über *STEP 7* oder SFC 28
Verzögerungsalarm	OB 20	• Ablauf der in SFC 32 angegebenen Zeit
Weckalarm (Default: 100 ms-Takt)	OB 35	• je nach Zeittakt (Parametrierung mit *STEP 7*)
Prozessalarm	OB 40	• Prozessalarm wurde ausgelöst
Diagnosealarm	OB 82	• gehendes Ereignis (Ereignisursache steht nicht mehr an) • kommendes Ereignis (Ereignisursache steht noch an)
Ziehen/Stecken-Alarm	OB 83	• Ziehen und Stecken von Baugruppen im Betriebszustand RUN

Bei fehlenden OB 10, 20, 35, 40, 82 oder 83 und entsprechendem Startereignis geht das IM 151-7 CPU in den Betriebszustand STOP.

OBs für Fehlerreaktionen

Tabelle 7-16 OBs für Fehlerreaktionen

Fehler	aufgerufener OB	mögliche Startereignisse
Zeitfehler	OB 80	• Überschreiten der Zykluszeit • Quittierungsfehler bei Bearbeiten eines OB • Vorstellen der Uhr (Zeitsprung) zum Starten eines OB
Programmablauf-fehler	OB 85	• Startereignis für einen nicht geladenen OB • Peripheriezugriffsfehler bei der systemseitigen Aktualisierung des Prozessabbildes
Ausfall/Wiederkehr des DP-Masters oder eines Teilnehmers des direkten Datenaustausches	OB 86	• gehendes Ereignis (Ereignisursache steht nicht mehr an) • kommendes Ereignis (Ereignisursache steht noch an)
Kommunikations-fehler	OB 87	• falsche Telegrammkennung bei GD • GD-Paketstatus nicht in DB eintragbar • GD-Gesamtstatus nicht in DB eintragbar
Programmierfehler	OB 121	• durch einen Fehler bei der Programmbearbeitung auftretendes Ereignis, z. B. durch Aufruf eines nicht in die CPU geladenen Bausteins
Fehler bei Peripherie-Direktzugriff	OB 122	• Fehler bei Lesezugriff • Fehler bei Schreibzugriff

Bei fehlenden OB 80, 85, 86, 87, 121 oder OB 122 und entsprechendem Startereignis geht das IM 151-7 CPU in den Betriebszustand STOP.

Besonderheiten des OB 122

Hinweis

Beachten Sie zum OB 122 folgende Besonderheit:

Der CPU-Teil trägt in die Lokaldaten des OBs in folgende temporäre Variablen der Variablendeklarationstabelle den Wert “0” ein:

- **Byte Nr. 3:** OB122_BLK_TYPE
  (Art des Bausteins, in dem der Fehler aufgetreten ist)
- **Byte Nr. 8 und 9:** OB122_BLK_NUM
  (Nummer des Bausteins, in dem der Fehler aufgetreten ist)
- **Byte Nr. 10 und 11:** OB122_PRG_ADDR
  (Adresse des Bausteins, in dem der Fehler aufgetreten ist)

7.9 Parameter

Parametrierbare Eigenschaften des CPU-Teils

Die Eigenschaften und das Verhalten des CPU-Teils des IM 151-7 CPU lassen sich parametrieren. Die Parametrierung nehmen Sie in *STEP 7* in verschiedenen Registern vor.

Welche Parameter sind für IM 151-7 CPU einstellbar?

Die folgende Tabelle enthält alle Parameterblöcke für das IM 151-7 CPU. Die Erläuterung der Parameter finden Sie in der Online-Hilfe *STEP 7*.

Tabelle 7-17 Parameterblöcke, einstellbare Parameter und ihre Wertebereiche für IM 151-7 CPU

Parameterblöcke	einstellbare Parameter	Wertebereich
Taktmerker	Taktmerker	ja/nein
	Merkerbyte	von 0 bis 255
Anlaufverhalten	Automatischer/Manueller Anlauf nach “NETZ-EIN”	Neustart
	Überwachungszeit für : • Fertigmeldung durch Baugruppen (100 ms) • Übertragung der Parameter an Baugruppen (100 ms)	 • von 1 bis 65000 • von 1 bis 65000
	Anlauf bei Sollausbau ungleich Istausbau	ja/nein
Systemdiagnose	STOP-Ursache melden	ja/nein
Remanenz	Anzahl Merkerbytes ab MB 0	0 bis 255
	Anzahl S7-Timer ab T 0	0 bis 255
	Anzahl S7-Zähler ab Z 0	0 bis 255
Hardware-Uhr	Korrekturfaktor	von –10000 bis +10000
Uhrzeitalarme	Aktivierung OB 10	ja/nein
	Ausführung OB 10	• keine • einmalig • minütlich • stündlich • täglich • wöchentlich • monatlich • monatsletzter • jährlich
	Startdatum OB 10	Jahr-Monat-Tag
	Startzeit OB 10	Stunden:Minuten
Weckalarme	Periodizität des OB 35 (ms)	von 1 bis 60000

Tabelle 7-17 Parameterblöcke, einstellbare Parameter und ihre Wertebereiche für IM 151-7 CPU

Parameterblöcke	einstellbare Parameter	Wertebereich
Zyklusverhalten	Zyklusbelastung durch Kommunikation (%)	von 10 bis 50
	Zyklusüberwachungszeit (ms)	von 1 bis 6000
	OB 85-Aufruf bei Peripheriezugriffsfehler	• bei jedem Zugriff • bei kommenden und gehenden Fehlern • kein Aufruf
Schutz	Schutzstufe	• 1: Schlüsselschalter • 2: Schreibschutz • 3: Schreib-/Leseschutz
	Betrieb	• Prozessbetrieb: zulässige Zykluszeiterhöhung (ms) von 3 bis 65535 • Testbetrieb
Baugruppenparameter	Anzahl Vergleichsstellen • Aktiviert • Modulnummer • Kanalnummer	1 • ja/nein • von 5 bis 66 • 0/1
	Störfrequenzunterdrückung	50 Hz / 60 Hz
DP-Teilnehmeradresse	CPU-DP-Adresse	von 1 bis 125

Wann “übernimmt” der CPU-Teil die Parameter?

Der CPU-Teil übernimmt die eingestellten Parameter (Konfigurationsdaten):

- nach NETZ-EIN oder Urlöschen vom gesteckten Speichermodul
- nachdem die Konfigurationsdaten im Betriebszustand STOP online zum CPU-Teil fehlerfrei übertragen worden sind.

7.10 Parametrierung der Vergleichsstelle beim Anschluss von Thermoelementen

Wenn Sie das IM 151-7 CPU in einem ET 200S-System mit Thermoelementen und Vergleichsstellen verwenden wollen, stellen Sie in der Hardware-Konfiguration folgende Parameter ein:

Tabelle 7-18 Parametrierung der Vergleichsstelle

CPU-Baugruppenparameter	Wertebereich	Erklärung
Aktivierung der Vergleichsstelle 1	Aktiviert/nicht aktiviert Beispiel siehe Bild 7-8	Mit diesem Parameter können Sie die Vergleichsstelle frei schalten. Erst dann können Sie die Vergleichsstelle weiter parametrieren.
Modulnummer der Vergleichsstelle 1 Die Modulnummer entspricht dem Steckplatz.	keinen/5 bis 66 Beispiel siehe Bild 7-8	Mit diesem Parameter können Sie den Steckplatz des RTD-Moduls der entsprechenden Vergleichsstelle zuordnen.
Kanalnummer der Vergleichsstelle 1	RTD an Kanal 0 RTD an Kanal 1 Beispiel siehe Bild 7-8	Mit diesem Parameter legen Sie den Kanal (0/1) zur Vergleichstemperaturmessung (Ermittlung des Kompensationswertes) für den zugeordneten RTD-Modul-Steckplatz fest.
RTD-Modul-Parameter	**Wertebereich**	**Erklärung**
Messart/Messbereich	Widerstands-/Temperaturmesssung, z. B. • RTD-4L Pt 100 Standardbereich	Wenn Sie einen Kanal des RTD-Moduls für eine Vergleichsstellenparametrierung verwenden, müssen Sie die Messart/den Messbereich für diesen Kanal als *RTD-4L Pt 100 Klimabereich* parametrieren.
TC-Modul-Parameter	**Wertebereich**	**Erklärung**
Vergleichsstellennummer	1	Mit diesem Parameter weisen Sie die Vergleichsstelle (1) zu, die die Vergleichstemperatur (Kompensationswert) enthält.
Vergleichsstelle Kanal 0 und Vergleichsstelle Kanal 1	keine, RTD	Über diesen Parameter können Sie die Verwendung der Vergleichsstelle freigeben.

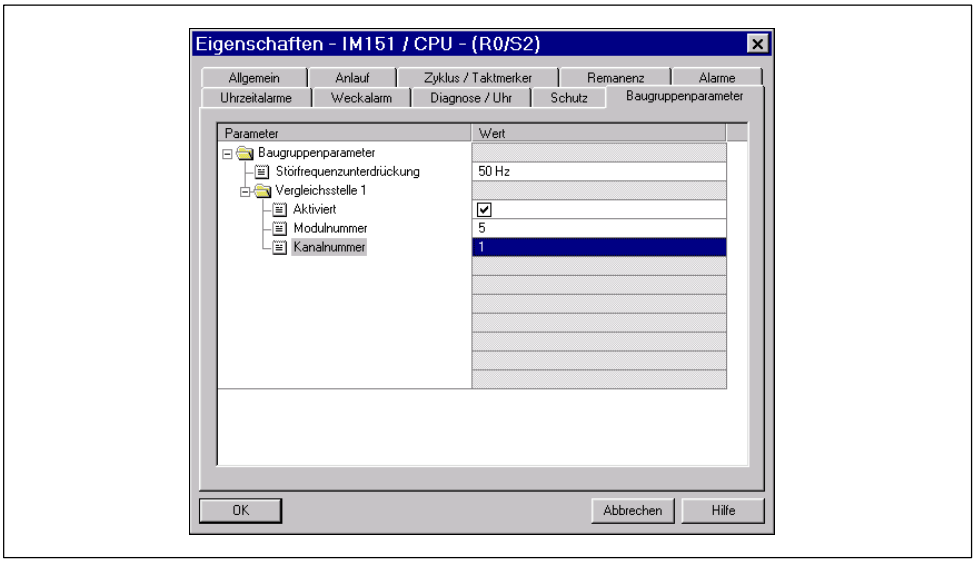


Bild 7-8 Beispiel für ein Parametrierfenster der CPU-Baugruppendaten in STEP 7 V5.1 + SP4

Verweis

Detaillierte Informationen zum Verfahren, zur Anschlusstechnik und ein Parametrierbeispiel finden Sie im Handbuch *Dezentrales Peripheriegerät ET 200S* im Kapitel *Analoge Elektronikmodule*.

7.11 Ziehen und Stecken von Modulen im laufenden Betrieb

Ziehen und Stecken von jeweils einem Modul beim IM 151-7 CPU mit lokaler ET 200S-Peripherie ist im laufenden Betrieb und unter Spannung erlaubt.

Ausnahme

Die CPU selbst darf im laufenden Betrieb und unter Spannung nicht gezogen werden.

Ziehen und Stecken von Modulen unter Spannung und im laufenden Betrieb

Beachten Sie beim Ziehen und Stecken von Modulen unter Spannung und im laufenden Betrieb neben den Angaben im Handbuch auch die Einschränkungen im Handbuch *Dezentrales Peripheriegerät ET 200S, Kapitel: "Verdrahten und Bestücken"*.

Warnung

Beim Stecken eines Ausgabemoduls sind die durch das Anwenderprogramm gesetzten Ausgänge sofort aktiv. Wir empfehlen Ihnen deshalb, schon vor dem Ziehen des Moduls im Anwenderprogramm die Ausgänge auf "0" zu setzen.

Bei nicht ordnungsgemäßem Ziehen und Stecken von Modulen (siehe Handbuch: *Dezentrales Peripheriegerät ET 200S, Kapitel: "Verdrahten und Bestücken"*) kann es zu unkontrollierten Anlagenzuständen kommen. Benachbarte Module können gestört werden.

Ablauf beim Ziehen von Modulen im laufenden Betrieb

Wenn Sie bei der ET 200S-Peripherie im laufenden Betrieb ein Modul ziehen, wird unabhängig davon, ob das Powermodul ein- oder ausgeschaltet ist, der OB 83 aufgerufen und ein entsprechender Diagnosepuffereintrag erzeugt (Ereignis-ID 3861_H).

Wenn der OB 83 im Speicher der CPU vorhanden ist, bleibt das IM 151-7 CPU im RUN.

Das Fehlen des Modules wird in der Systemzustandsliste vermerkt.

Wenn aus dem Anwenderprogramm heraus ein Zugriff auf das gezogene Modul erfolgt, tritt eine Peripheriezugriffsfehler mit entsprechendem Diagnosepuffereintrag auf. Außerdem wird der OB 122 aufgerufen.

Wenn der OB 122 im Speicher der CPU vorhanden ist, bleibt das IM 151-7 CPU im RUN.

Ablauf beim Stecken von Modulen im laufenden Betrieb

Wenn Sie bei der ET 200S-Peripherie im laufenden Betrieb ein gezogenes Modul wieder stecken, führt die CPU zunächst einen Soll-Ist-Vergleich bezüglich des gesteckten Moduls durch. Hierbei wird das projektierte mit dem tatsächlich gesteckten Modul verglichen. Abhängig vom Ergebnis des Soll-Ist-Vergleiches finden folgende Aktivitäten statt:

Nicht parametrierbare Module

Folgende Aktivitäten finden unabhängig davon statt, ob das Powermodul des gesteckten Modules ein- oder ausgeschaltet ist.

Tabelle 7-19 Ergebnis des Soll-Ist-Vergleiches bei **nicht** parametrierbaren Modulen

Gestecktes Modul = Projektiertes Modul	**Gestecktes Modul ≠ Projektiertes Modul**
Aufruf des OB 83 mit entsprechendem Diagnosepuffereintrag (Ereignis-ID 3861_H).	Aufruf des OB 83 mit entsprechendem Diagnosepuffereintrag (Ereignis-ID 3863_H).
Das Modul wird in der Systemzustandsliste als verfügbar eingetragen.	Das Modul bleibt in der Systemzustandsliste als nicht verfügbar eingetragen.
Direktzugriffe sind damit wieder möglich.	Direktzugriffe sind nicht möglich.

Parametrierbare Module

Folgende Aktivitäten finden nur dann statt, wenn das Powermodul des gesteckten Modules **ein**geschaltet ist.

Tabelle 7-20 Ergebnis des Soll-Ist-Vergleiches bei parametrierbaren Modulen, Powermodul eingeschaltet

Gestecktes Modul = Projektiertes Modul	**Gestecktes Modul ≠ Projektiertes Modul**
Aufruf des OB 83 mit entsprechendem Diagnosepuffereintrag (Ereignis-ID 3861_H).	Aufruf des OB 83 mit entsprechendem Diagnosepuffereintrag (Ereignis-ID 3863_H).
Die CPU parametriert das Modul neu.	Die CPU parametriert das Modul nicht.
Wenn die Parametrierung erfolgreich ist, wird das Modul in der Systemzustandsliste als verfügbar eingetragen.	Das Modul bleibt in der Systemzustandsliste als nicht verfügbar eingetragen. Die SF-LED am Modul leuchtet weiterhin.
Direktzugriffe sind damit wieder möglich.	Direktzugriffe sind nicht möglich.

Folgende Aktivitäten finden nur dann statt, wenn das Powermodul des gesteckten Modules **aus**geschaltet ist.

Tabelle 7-21 Ergebnis des Soll-Ist-Vergleiches bei parametrierbaren Modulen, Powermodul ausgeschaltet

Gestecktes Modul = Projektiertes Modul	**Gestecktes Modul ≠ Projektiertes Modul**
Aufruf des OB 83 mit entsprechendem Diagnosepuffereintrag (Ereignis-ID 3861_H).	
Wenn das Powermodul eingeschaltet wird, parametriert die CPU das Modul neu.	Wenn das Powermodul eingeschaltet wird, parametriert die CPU das Modul nicht.
Wenn die Parametrierung erfolgreich ist, wird das Modul in der Systemzustandsliste als verfügbar eingetragen.	Das Modul bleibt in der Systemzustandsliste als nicht verfügbar eingetragen. Die SF-LED am Modul leuchtet weiterhin.
Direktzugriffe sind damit wieder möglich.	Direktzugriffe sind nicht möglich.

7.12 Aus- und Einschalten von Powermodulen im laufenden Betrieb

Ablauf beim Ausschalten von Powermodulen im laufenden Betrieb

Wird im laufenden Betrieb an einem Powermodul die Lastspannungversorgung abgeschaltet, so finden folgende Aktivitäten statt:

- Wenn Sie die Diagnose in der Parametrierung für das Powermodul freigegeben haben, wird der Diagnosealarm-OB 82 (Diagnoseadresse des Powermoduls) mit entsprechendem Diagnosepuffereintrag aufgerufen (Ereignis 3942_H).
- In der Systemzustandsliste wird das Powermodul als vorhanden aber gestört eingetragen.

Für die Module, die durch das Powermodul versorgt werden, hat das Ausschalten der Lastspannungsversorgung folgende Auswirkungen:

- Die SF-LED der Module leuchtet.
- Auf die Module kann weiterhin zugegriffen werden, ein Peripheriezugriffsfehler ereignet sich nicht.
- Die Ausgänge der Module werden spannungslos und inaktiv zum Prozess.
- Die Eingänge von Digitalmodulen und FM-Modulen liefern 0, die Eingänge von Analogmodulen liefern $7FFF_{H.}$

Ablauf beim Einschalten von Powermodulen im laufenden Betrieb

Wird im laufenden Betrieb an einem Powermodul die Lastspannungversorgung eingeschaltet, so finden folgende Aktivitäten statt:

- Wenn Sie die Diagnose in der Parametrierung für das Powermodul freigegeben haben, wird der Diagnosealarm-OB 82 (Diagnoseadresse des Powermoduls) mit entsprechendem Diagnosepuffereintrag aufgerufen (Ereignis 3842_H).
- In der Systemzustandsliste wird das Powermodul als vorhanden und o.k. eingetragen.

Für die Module, die durch das Powermodul versorgt werden, hat das Einschalten der Lastspannungsversorgung folgende Auswirkungen:

- Die SF-LED der Module verlischt.
- Die Module erhalten Ihre volle Funktionalität zurück.

Ziehen und Stecken von Powermodulen im laufenden Betrieb

Wenn Sie im laufenden Betrieb ein Powermodul ziehen oder stecken, finden die im Kapitel 7.11 dargestellten Aktivitäten statt.

Für die Module, die durch das Powermodul versorgt werden, hat das Ziehen oder Stecken die gleichen Auswirkungen wie das Aus- oder Einschalten der Lastspannungsversorgung.

Zyklus- und Reaktionszeiten

8

Einleitung

In diesem Kapitel erfahren Sie, wie sich die Zyklus- und Reaktionszeiten des ET 200S mit IM 151-7 CPU zusammensetzen.

Die Zykluszeit Ihres Anwenderprogramms können Sie mit dem PG auslesen (siehe *Benutzerhandbuch STEP 7*).

Wichtiger für die Betrachtung eines Prozesses ist die Reaktionszeit. Wie Sie diese berechnen, zeigen wir Ihnen ausführlich in diesem Kapitel.

Kapitelübersicht

Kapitel	Thema	Seite
8.1	Zykluszeit	8-2
8.2	Reaktionszeit	8-5
8.3	Alarmreaktionszeit	8-8

Ausführungszeiten

- für die von den CPUs verarbeitbaren *STEP 7*-Anweisungen finden Sie im Anhang A.
- für die in den CPUs integrierten SFCs/SFBs finden Sie im Anhang B.

8.1 Zykluszeit

Definition Zykluszeit

Die Zykluszeit ist die Zeit, die das Betriebssystem für die Bearbeitung eines Programmdurchlaufes – d.h. eines OB 1-Durchlaufes – sowie aller diesen Durchlauf unterbrechenden Programmteile und Systemtätigkeiten benötigt.

Diese Zeit wird überwacht.

Teile der Zykluszeit

Faktoren	Bemerkung
Betriebssystembearbeitungszeit	siehe Tabelle 8-1
Prozeßabbildtransferzeit (PAE und PAA)	siehe Tabelle 8-2
Anwenderprogrammbearbeitungszeit	... errechnen Sie aus den Ausführungszeiten der einzelnen Operationen und einem CPU-spezifischen Faktor (siehe Tabelle 8-3).
Belastung durch Alarme	siehe Tabelle 8-4

Das folgende Bild zeigt die Teile der Zykluszeit.

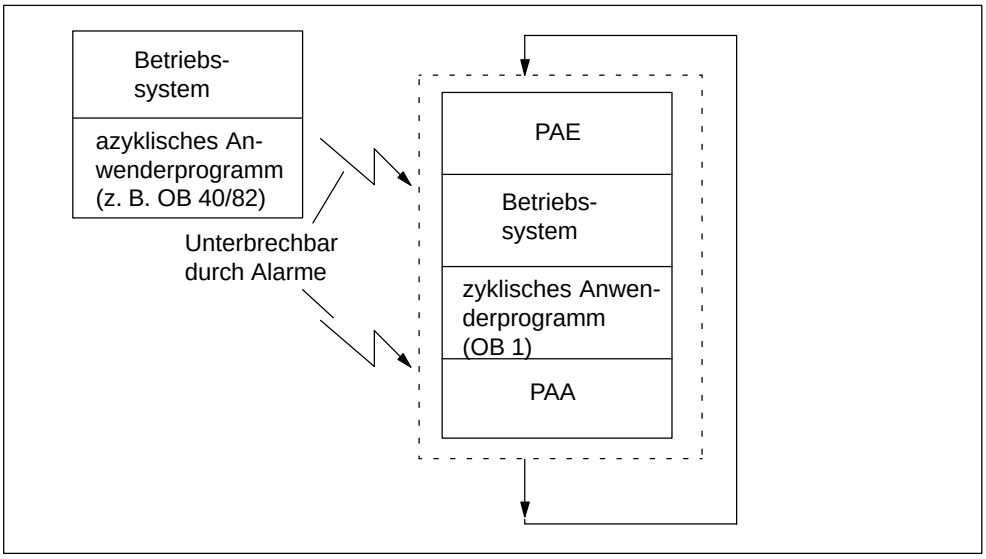


Bild 8-1 Teile der Zykluszeit

Verlängerung der Zykluszeit

Prinzipiell müssen Sie beachten, daß sich die Zykluszeit eines Anwenderprogramms verlängert durch:

- zeitgesteuerte Alarmbearbeitung
- Prozeßalarmbearbeitung (siehe auch Kapitel 8.3)
- Diagnose und Fehlerbearbeitung (siehe auch Kapitel 8.3)

Betriebssystembearbeitungszeit

Die Betriebssystembearbeitungszeit beträgt beim IM 151-7 CPU 800 µs (siehe Tabelle 8-1).

Die angegebene Zeit gilt ohne Ausführung von:

- Testfunktionen, zum Beispiel Status, Steuern
- Funktionen Baustein-Laden, -Löschen, -Komprimieren
- Kommunikation
- Beschreiben, Lesen der MMC mit SFC 82 bis 84

Tabelle 8-1 Betriebssystembearbeitungszeit im Zykluskontrollpunkt

Ablauf	IM 151-7 CPU
Betriebssystembearbeitungszeit	600 µs

Prozessabbild-Transferzeit

Die folgende Tabelle enthält die CPU-Zeiten für die Prozessabbild-Aktualisierung (Prozessabbild-Transferzeit). Die angegebenen Zeiten können sich durch auftretende Alarme oder durch Kommunikation des CPU-Teils des IM 151-7 CPU verlängern.

(Prozessabbild = PA)

Die CPU-Zeit für die Prozessabbild-Aktualisierung berechnet sich nach:

K + A + D = Prozessabbild-Transferzeit, wobei

Tabelle 8-2 Prozessabbild-Aktualisierung

	Benennung	Zeiten in IM 151-7 CPU
K	Grundlast	100 µs
A	Bytes im PA für ET 200S-Peripherie	60 µs je Byte
D	je Wort im DP-Bereich	1 µs

Anwenderprogrammbearbeitungszeit

Die Anwenderprogrammbearbeitungszeit setzt sich zusammen aus der Summe der Ausführungszeiten der Operationen und der aufgerufenen SFCs. Diese Ausführungszeiten finden Sie in der Operationsliste. Zusätzlich müssen Sie die Anwenderprogrammbearbeitungszeit mit einem Basismodul-spezifischen Faktor multiplizieren.

Dieser Faktor ist beim IM 151-7 CPU abhängig von:

Tabelle 8-3 Abhängigkeit der Anwenderprogrammbearbeitungszeit

Abhängigkeit	Wertebereich
Anzahl der gesteckten Module	0 bis 63

Den Faktor für Ihre Applikation können Sie anhand nachstehender Faustformel für das IM 151-7 CPU näherungsweise berechnen:

1,1
\+ 0,005 x Anzahl der Module
= Multiplikator für Ihr Anwenderprogramm

Verzögerung der Ein-/Ausgänge

Sie müssen je nach Erweiterungsmodul folgende Verzögerungszeiten beachten:

- für Digitaleingänge: die Eingangsverzögerungszeit
- für Digitalausgänge: vernachlässigbare Verzögerungszeiten
- für Analogeingänge: Zykluszeit der Analogeingabe
- für Analogausgänge: Antwortzeit der Analogausgabe

Zyklusverlängerung durch Einschachtelung von Alarmen

Tabelle 8-4 zeigt, wie sich die Zykluszeit durch das Einschachteln eines Alarms typisch verlängert. Zu dieser Verlängerung kommt die Programmlaufzeit in der Alarmebene hinzu. Werden mehrere Alarme eingeschachtelt, dann addieren sich die entsprechenden Zeiten.

Tabelle 8-4 Zyklusverlängerung durch Einschachteln von Alamen

Alarme	IM 151-7 CPU
Prozeßalarm	500 µs
Diagnosealarm	600 µs
Uhrzeitalarm	400 µs
Verzögerungsalarm	300 µs
Weckalarm	150 µs
Programmier-/Zugriffsfehler/Programmablauffehler	400 µs

8.2 Reaktionszeit

Reaktionszeit für ET 200S mit IM 151-7 CPU

Die Reaktionszeit ist die Zeit vom Erkennen eines Eingangssignals an ET 200S mit IM 151-7 CPU bis zur Änderung eines damit verknüpften Ausgangssignals über die Ein- und Ausgänge der Erweiterungsmodule.

Faktoren

Die Reaktionszeit hängt von der Zykluszeit und von folgenden Faktoren ab:

Faktoren	Bemerkung
Verzögerung der Eingänge und Ausgänge	Die Verzögerungszeiten finden Sie in den technischen Daten der Elektronikmodule im Handbuch *Dezentrales Peripheriegerät ET 200S*.

Schwankungsbreite

Die tatsächliche Reaktionszeit liegt zwischen einer kürzesten und einer längsten Reaktionszeit. Zur Projektierung Ihrer Anlage müssen Sie immer mit der längsten Reaktionszeit rechnen.

Im Folgenden werden kürzeste und längste Reaktionszeit betrachtet, damit Sie sich ein Bild von der Schwankungsbreite der Reaktionszeit machen können.

Kürzeste Reaktionszeit

Das folgende Bild zeigt Ihnen, unter welchen Bedingungen die kürzeste Reaktionszeit erreicht wird.

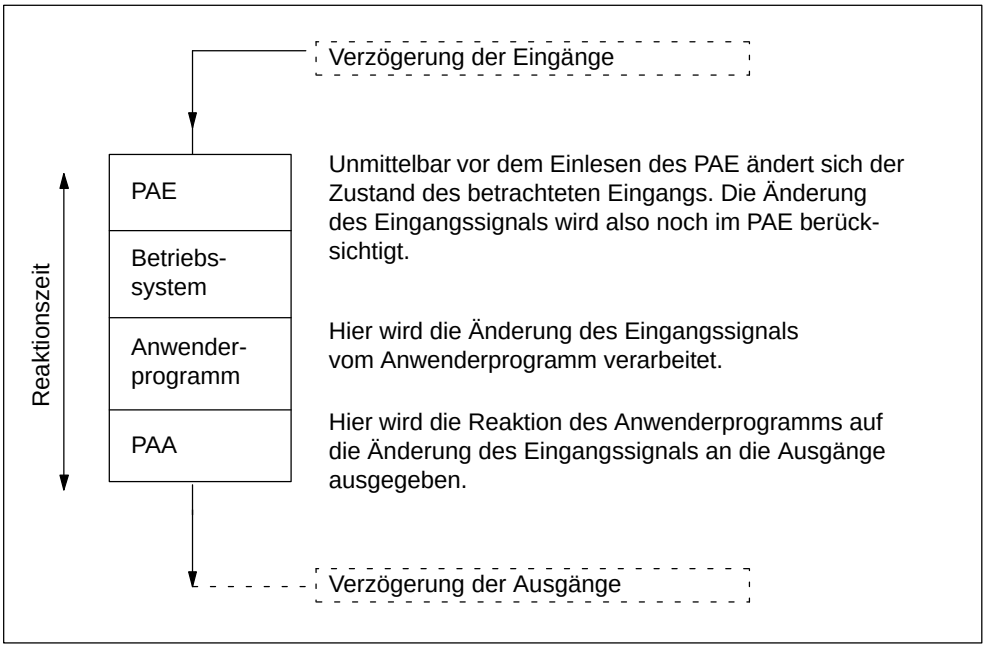

Verzögerung der Eingänge
PAE
Betriebs-
system
Anwender-
programm
PAA
Reaktionszeit
Unmittelbar vor dem Einlesen des PAE ändert sich der Zustand des betrachteten Eingangs. Die Änderung des Eingangssignals wird also noch im PAE berücksichtigt.
Hier wird die Änderung des Eingangssignals vom Anwenderprogramm verarbeitet.
Hier wird die Reaktion des Anwenderprogramms auf die Änderung des Eingangssignals an die Ausgänge ausgegeben.
Verzögerung der Ausgänge

Bild 8-2 Kürzeste Reaktionszeit

Berechnung

Die (kürzeste) Reaktionszeit setzt sich wie folgt zusammen:

- 1 × Prozessabbild-Transferzeit der Eingänge +
- 1 × Betriebssystembearbeitungszeit +
- 1 × Programmbearbeitungszeit +
- 1 × Prozessabbild-Transferzeit der Ausgänge +
- Verzögerung der Eingänge und Ausgänge

Dies entspricht der Summe aus Zykluszeit und Verzögerung der Eingänge und Ausgänge.

Längste Reaktionszeit

Das folgende Bild zeigt Ihnen, wodurch die längste Reaktionszeit zustande kommt.

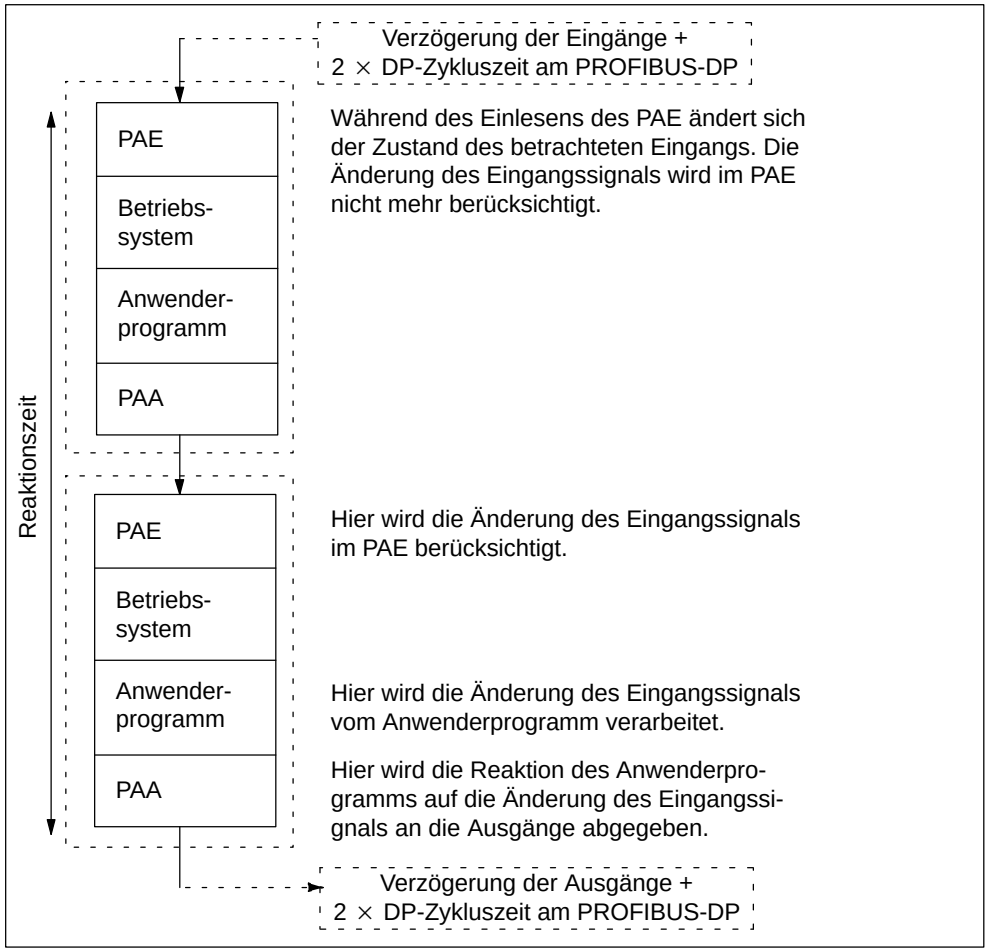

Verzögerung der Eingänge +
2 × DP-Zykluszeit am PROFIBUS-DP
PAE
Betriebs-
system
Anwender-
programm
PAA
Während des Einlesens des PAE ändert sich der Zustand des betrachteten Eingangs. Die Änderung des Eingangssignals wird im PAE nicht mehr berücksichtigt.
Reaktionszeit
PAE
Betriebs-
system
Anwender-
programm
PAA
Hier wird die Änderung des Eingangssignals im PAE berücksichtigt.
Hier wird die Änderung des Eingangssignals vom Anwenderprogramm verarbeitet.
Hier wird die Reaktion des Anwenderprogramms auf die Änderung des Eingangssignals an die Ausgänge abgegeben.
Verzögerung der Ausgänge +
2 × DP-Zykluszeit am PROFIBUS-DP

Bild 8-3 Längste Reaktionszeit

Berechnung

Die (längste) Reaktionszeit setzt sich wie folgt zusammen:

- 2 × Prozessabbild-Transferzeit der Eingänge +
- 2 × Prozessabbild-Transferzeit der Ausgänge +
- 2 × Betriebssystembearbeitungszeit +
- 2 × Programmbearbeitungszeit +
- 4 × Laufzeit des DP-Slavetelegramms (inkl. Bearbeitung im DP-Master) +
- Verzögerung der Eingänge und Ausgänge

Dies entspricht der Summe aus doppelter Zykluszeit und Verzögerung der Eingänge und Ausgänge zuzüglich der vierfachen DP-Zykluszeit.

8.3 Alarmreaktionszeit

Definition Alarmreaktionszeit

Die Alarmreaktionszeit ist die Zeit vom ersten Auftreten eines Alarmsignals bis zum Aufruf der ersten Anweisung im Alarm-OB des IM 151-7 CPU.

Generell gilt: Höherpriore Alarme haben Vorrang. Das heißt, die Alarmreaktionszeit verlängert sich um die Programmbearbeitungszeit des höherprioren und der noch nicht bearbeiteten gleichprioren vorher aufgetretenen Alarm-OBs (Warteschlange).

Alarmreaktionszeiten

Tabelle 8-5 Alarmreaktionszeiten des IM 151-7 CPU (ohne Kommunikation)

Alarmreaktionszeiten (ohne Kommunikation) bei ...	Dauer
Prozessalarm, Diagnosealarm	weniger als 20 ms

Prozessalarmbearbeitung

Mit dem Aufruf des Prozessalarm-OB 40 erfolgt die Prozessalarmbearbeitung. Höherpriore Alarme unterbrechen die Prozessalarmbearbeitung, Direktzugriffe auf die Peripherie erfolgen zur Ausführungszeit der Anweisung. Nach Beendigung der Prozessalarmbearbeitung wird entweder die zyklische Programmbearbeitung fortgesetzt oder weitere gleichpriore bzw. niederpriore Alarm-OBs werden aufgerufen und bearbeitet.

Technische Daten

9

In diesem Kapitel

Sie finden in diesem Kapitel:

- die technischen Daten des Interfacemoduls IM 151-7 CPU

Kapitelübersicht

Kapitel	Thema	Seite
9.1	Technische Daten des IM 151-7 CPU	9-2

9.1 Technische Daten des IM 151-7 CPU

Bestellnummern

Interfacemodul IM 151-7 CPU:	6ES7 151-7AA10-0AB0
Interfacemodul IM 151-7 CPU FO:	6ES7 151-7AB10-0AB0
SIMATIC Micro Memory Card MMC: (siehe Kapitel 7.3)	6ES7 953-8Lxx0-0AA0

Eigenschaften

Das Interfacemodul IM 151-7 CPU verfügt über folgende Eigenschaften:

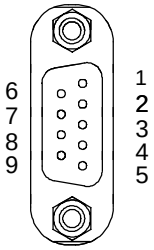
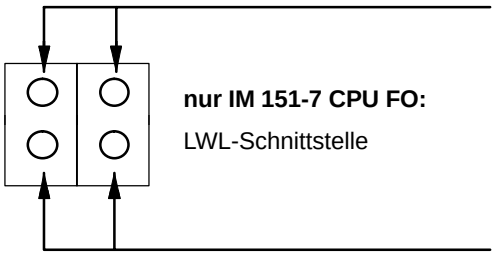
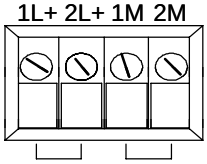
- intelligenter Slave mit RS485-Schnittstelle oder LWL an PROFIBUS-DP
- Stand-alone-Betrieb (MPI) möglich
- 48 kByte Arbeitsspeicher, nicht erweiterbar, remanent mit gesteckter MMC
- steckbarer Ladespeicher auf der MMC, bis zu 8 MByte
- netzausfallsichere Speicherung von Anwenderprogramm und Projektierung über MMC
- programmierbar mit *STEP 7*, ab V5.1 + Service Pack 4
- Geschwindigkeit: ca. 0,3 ms pro 1000 Binäranweisungen
- maximaler Ausbau der lokalen Peripherie: 63 ET 200S-Module

Allgemeine Technische Daten

Das IM 151-7 CPU erfüllt die allgemeinen technischen Daten des Dezentralen Peripheriesystems ET 200S. Diese Normen und Prüfwerte finden Sie im Kapitel “Allgemeine technische Daten” im Handbuch *Dezentrales Peripheriesystem ET 200S*.

Anschlussbelegung für IM 151-7 CPU

Tabelle 9-1 Anschlussbelegung des Interfacemoduls IM 151-7 CPU

Ansicht	Signalname		Bezeichnung
IM 151-7 CPU  RS 485-Schnittstelle	1	–	–
	2	M24	externe 24 V-Versorgung
	3	RxD/TxD-P	Datenleitung-B
	4	RTS	Request To Send
	5	M5V2	Datenbezugspotential (von Station)
	6	P5V2	Versorgungs-Plus (von Station)
	7	P24	externe 24 V-Versorgung
	8	RxD/TxD-N	Datenleitung-A
	9	–	–
 **nur IM 151-7 CPU FO:** LWL-Schnittstelle			
1L+ 2L+ 1M 2M 	1L+		DC 24 V
	2L+		DC 24 V (zum Weiterschleifen)
	1M		Masse
	2M		Masse (zum Weiterschleifen)

Prinzipschaltbild für IM 151-7 CPU

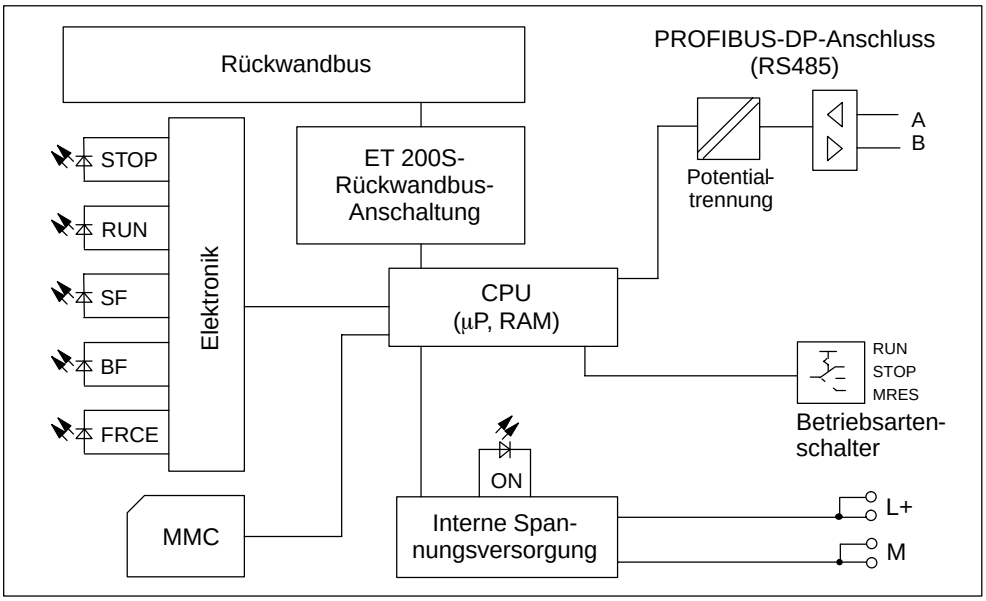


Bild 9-1 Prinzipschaltbild IM 151-7 CPU

Prinzipschaltbild für IM 151-7 CPU FO

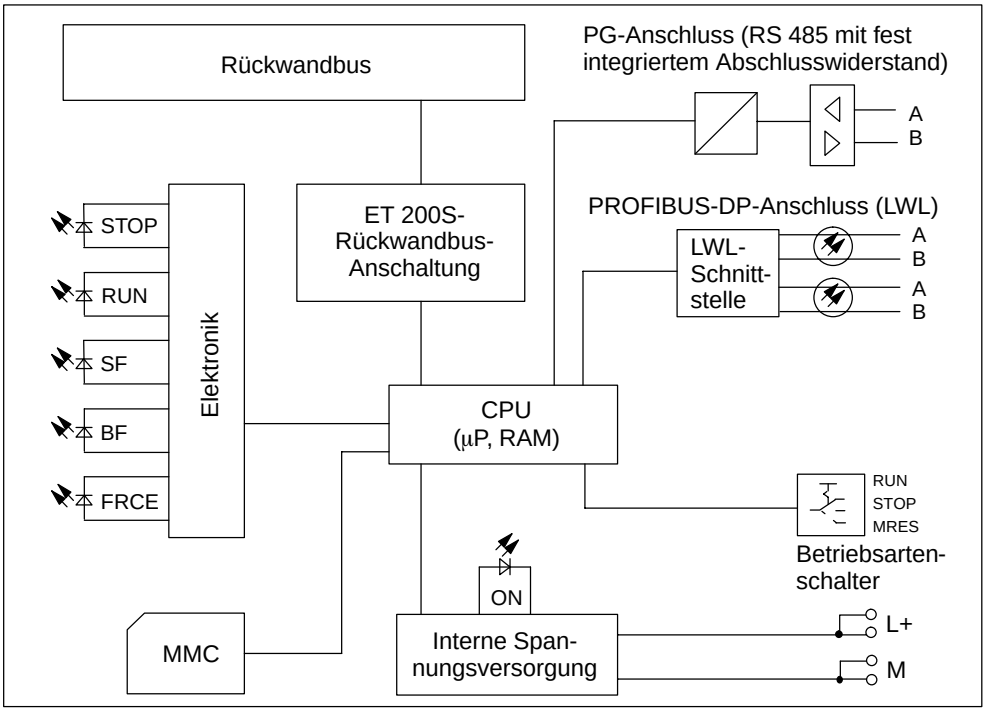


Bild 9-2 Prinzipschaltbild IM 151-7 CPU FO

Technische Daten

CPU und Erzeugnisstand	
MLFB	6ES7 151-7AA10-0AB0 FO: 6ES7 151-7AB10-0AB0
• Hardware-Erzeugnissstand	1
• Firmware-Erzeugnisstand	V2.0.0
• zugehöriges Programmierpaket	STEP 7 ab V5.1 + SP4
Speicher	
Arbeitsspeicher:	
• integriert	48 kByte
• erweiterbar	nein
Ladespeicher:	steckbar (MMC bis 8 MB)
Pufferung	durch MMC gewährleistet (wartungsfrei)
Bearbeitungszeiten	
Bearbeitungszeiten für	
• Bitoperationen	min. 0,1 µs
• Wortoperationen	min. 1 µs
• Festpunktarithmetik	min. 2 µs
• Gleitpunktarithmetik	min. 20 µs
Zeiten, Zähler und deren Remanenz	
S7-Zähler	256
• Remanenz	einstellbar
• voreingestellt	von Z 0 bis Z 7
• Zählbereich	0 bis 999
IEC-Counter	ja
• Art	SFB
S7-Zeiten	256
• Remanenz	einstellbar
• voreingestellt	keine Zeiten remanent
• Zeitbereich	10 ms bis 9990 s
IEC-Timer	ja
• Art	SFB

Datenbereiche und deren Remanenz	
remanenter Datenbereich gesamt (inkl. Merker, Zeiten, Zähler)	alle
Merker	256 Byte
• Remanenz	einstellbar
• voreingestellt	von MB 0 bis MB 15
Taktmerker	8 (1 Merkerbyte)
Datenbausteine	max. 511 (DB 0 reserviert)
• Größe	max. 16 kByte
Lokaldaten je Prioritätsklasse	max. 510 Byte
Bausteine	
OBs	siehe Kapitel 7.8
• Größe	max. 16 kByte
Schachtelungstiefe:	
• je Prioritätsklasse	8
• zusätzliche innerhalb eines Fehler-OBs	4
FBs	max. 512
• Größe	max. 16 kByte
FCs	max. 512
• Größe	max. 16 kByte
Adressbereiche (Ein-/Ausgänge)	
Peripherieadressbereich gesamt	max. 2048 Byte/2048 Byte
Prozessabbild	128 Byte/128 Byte (nicht einstellbar)
digitale Kanäle	max. 248/248
analoge Kanäle	max. 124/124

Uhrzeit	
Uhr	Hardware-Uhr
• gepuffert	ja
• Pufferungsdauer	typ. 6 Wochen (bei 40 °C Umgebungstemperatur)
• Genauigkeit	Abweichung pro Tag < 10 s
Betriebsstundenzähler	1
• Nummer	0
• Wertebereich	0 bis 32767 Stunden
• Granularität	1 Stunde
• remanent	ja; muss bei jedem Neustart neu gestartet werden

S7-Meldefunktionen	
Anzahl anmeldbarer Stationen für Meldefunktionen (z. B. OS)	max. 11
Prozessdiagnosemeldungen	ALARM_S, ALARM_SC, ALARM_SQ
• gleichzeitig aktive Alarm_S-Bausteine	max. 40

Test- und Inbetriebnahmefunktionen	
Status/Steuern Variable	ja
• Variable	Eingänge, Ausgänge, Merker, DB, Zeiten, Zähler
• Anzahl Variable	max. 30
– davon Status Variable	max. 30
– davon Steuern Variable	max. 14
Forcen	ja
• Variable	Eingänge, Ausgänge
• Anzahl	max. 10
Status Baustein	ja
Einzelschritt	ja
Haltepunkt	2
Diagnosepuffer	ja
• Anzahl der Einträge	max. 100 (nicht einstellbar)

Kommunikationsfunktionen	
PG-/OP-Kommunikation	ja
Globale Datenkommunikation	ja
• Anzahl der GD-Pakete	max. 4
– Sender	max. 4
– Empfänger	max. 4
• Größe der GD-Pakete	max. 22 Byte
– davon konsistent	22 Byte
S7-Basis-Kommunikation	Ja
• Nutzdaten pro Auftrag	max. 76 Byte
– davon konsistent	76 Byte (XSEND/XRECEIVE) 64 Byte (XPUT/XGET) als Server
S7-Kommunikation	Ja (Server)
• Nutzdaten pro Auftrag	max. 160 Byte
– davon konsistent	64 Byte
S5-kompatible-Kommunikation	nein
Standardkommunikation	nein
Anzahl Verbindungen	max. 12
verwendbar für	
• PG-Kommunikation	max. 11
– reserviert (Default)	1
• OP-Kommunikation	max. 11
– reserviert (Default)	1
• S7-Basis-Kommunikation	max. 8
– reserviert (Default)	0

Schnittstelle		**FO**
Typ der Schnittstelle	integrierte RS 485-Schnittstelle	LWL-Schnittstelle und integrierte PG-Schnittstelle
Physik	RS 485	LWL bzw. RS 485
potentialgetrennt	ja	nein
Stromversorgung an Schnittstelle (15 bis 30 V DC)	max. 80 mA	

Funktionalität	
• MPI	ja
• PROFIBUS-DP	DP-Slave (aktiv/passiv)
• Punkt-zu-Punkt-Kopplung	nein

MPI	
• Anzahl Verbindungen	12
• Dienste:	
– PG-/OP-Kommunikation	ja
– Routing	nein
– Globaldaten-Kommunikation	ja
– S7-Basiskommunikation	ja
– S7-Kommunikation	ja (nur Server)
• Übertragungsgeschwindigkeiten	max. 12 MBaud

DP-Slave	
• Anzahl Verbindungen	12
• Dienste:	
– PG-/OP-Kommunikation	ja
– Routing	nein
– Direkter Datenaustausch	ja
• GSD-Datei	siem80E2.gsg FO: siem80.. .gsg
• Übertragungsgeschwindigkeiten	bis 12 MBaud
• Übergabespeicher	244 Byte I/244 Byte O
– Adressbereiche	32 mit je max. 32 Byte*

Programmierung	
Programmiersprache	STEP 7 (KOP, FUP, AWL)
Operationsvorrat	siehe Anhang A
Klammerebenen	8
Systemfunktionen (SFC)	siehe Anhang B
Systemfunktionsbausteine (SFB)	siehe Anhang B
Anwenderprogrammschutz	ja

Maße und Gewicht	
Einbaumaße B × H × T (mm)	60 x 119,5 x 75
Gewicht	ca. 200 g

Spannungen, Ströme	
Versorgungsspannung Nennwert	DC 24 V
• zulässiger Bereich	20,4 bis 28,8 V
• Verpolschutz	ja
• Spannungsausfallüberbrückung	5 ms
Isolation geprüft mit	DC 500 V
Stromaufnahme aus Versorgungsspannung (1L+)	ca. 250 mA
• Stromversorgung für ET 200S Rückwandbus	max. 700 mA
Verlustleistung	typ. 3,3 W

* bis zur maximalen Größe des Übergabespeichers

Umstieg von IM 151-7 CPU (6ES7 151-7Ax00-0AB0) auf IM 151-7 CPU (6ES7 151-7Ax10-0AB0)

10

Wenn Sie Ihr vorhandenes Anwenderprogramm für das IM 151-7 CPU (6ES7 151-7Ax00-0AB0) auf ein IM 151-7 CPU (6ES7 151-7Ax10-0AB0) laden, kann es zu folgenden Problemen kommen:

Asynchron arbeitende SFC

Einige asynchron arbeitende SFC waren auf den IM 151-7 CPU (6ES7 151-7Ax00-0AB0) immer oder unter bestimmten Bedingungen bereits nach dem ersten Aufruf abgearbeitet (“quasi-synchron”).

Diese SFC laufen auf den IM 151-7 CPU (6ES7 151-7Ax10-0AB0) wirklich asynchron. Die asynchrone Bearbeitung kann sich über mehrere OB 1-Zyklen erstrekken. Dadurch kann eine Warteschleife innerhalb eines OB zu einer Endlosschleife werden.

Betroffen sind:

- SFC 56 “WR_DPARM”; SFC 57 “PARM_MOD”

  Dieser SFC arbeitet auf stand-alone IM 151-7 CPU (6ES7 151-7Ax00-0AB0) immer “quasi-synchron”.
  Auf stand-alone IM 151-7 CPU (6ES7 151-7Ax10-0AB0) und auf dezentralen IM 151-7 CPU arbeitet er asynchron.

Hinweis

Verwenden Sie den SFC 56 “WR_DPARM” oder SFC 57 “PARM_MOD”, sollten Sie immer das BUSY-Bit der SFCs auswerten.

Funktionseinschränkungen von SFC

SFC 20 “BLKMOV”

Dieser SFC konnte bisher auch verwendet werden, um Daten aus einem nicht ablaufrelevanten DB zu kopieren.

Diese Funktionalität hat der SFC 20 nicht mehr. Dafür ist jetzt der SFC 83 “READ_DBL” zu verwenden.

Nicht mehr verfügbare SFC

SFC 54 “RD_DPARM”

Dieser SFC ist nicht mehr verfügbar, stattdessen muss der asynchron arbeitende SFC 102 “RD_DPARA” verwendet werden.

SFC, die ggf. andere Ergebnisse liefern

Wenn Sie ausschließlich logische Adressierung in Ihrem Anwenderprogramm verwenden, brauchen Sie die folgenden Punkte nicht zu berücksichtigen.

Wenn Sie Adressumrechnungen im Anwenderprogramm verwenden (SFC 5 “GADR_LGC”, SFC 49 “LGC_GADR”), dann müssen Sie für DP-Slaves die Zuordnung von Steckplatz und logischer Anfangsadresse prüfen.

- Die Diagnoseadresse des DP-Slaves ist jetzt immer dem Steckplatz 0 zugeordnet.
- DP-Slave in *STEP 7* integriert:

  Die Anschaltungsbaugruppe (Steckplatz 2) hat ggf. eine eigene Adresse (z. B. IM 151-7 CPU als I-Slave).

Umstellung von Diagnoseadressen von DP-Slaves

Beachten Sie, dass Sie die Diagnoseadressen für die Slaves unter Umständen neu vergeben müssen, da zum Teil zwei Diagnoseadressen pro Slave erforderlich sind.

- Der virtuelle Steckplatz 0 hat eine eigene Adresse.

  Die Baugruppenzustandsdaten zu diesem Steckplatz (auslesen mit SFC 51 “RDSYSST”) enthalten die Kennungen, die den kompletten Slave/die komplette Station betreffen, z. B. Kennung Station gestört.

- Zusätzlich hat bei in *STEP 7* integrierten Baugruppen (z. B. IM 151-7 CPU als I-Slave) auch der Steckplatz 2 eine eigene Adresse. Über diese Adresse wird z. B. bei IM 151-7 CPU als I-Slave der Betriebszustandswechsel im Diagnosealarm OB 82 des Masters gemeldet.

Wenn Sie den SFC 51 “RDSYSST” benutzen, um z. B. Baugruppenzustandsinformation oder Baugruppenträger-/Stationszustandsinformation auszulesen, müssen Sie auch die geänderte Bedeutung der Steckplätze und den zusätzlichen Steckplatz 0 berücksichtigen.

Verwendung konsistenter Datenbereiche im Prozessabbild bei DP-Slaves

Nachfolgend zeigen wir Ihnen, was Sie bei der Kommunikation in einem DP-Mastersystem beachten müssen, wenn Sie E/A-Bereiche mit der Konsistenz “Gesamte Länge” übertragen wollen.

- Wenn der Adressbereich konsistenter Daten im Prozessabbild liegt, dann wird dieser Bereich automatisch aktualisiert.
- Zum Lesen und Schreiben konsistenter Daten können Sie auch die SFCs 14 und 15 benutzen.
- Wenn der Adressbereich konsistenter Daten außerhalb des Prozessabbilds liegt, dann müssen Sie zum Lesen und Schreiben konsistenter Daten die SFCs 14 und 15 benutzen.
- Außerdem sind auch Direktzugriffe auf die konsistenten Bereiche möglich (z. B. L PEW oder T PAW).

Sie können maximal 32 Byte konsistente Daten übertragen.

Ersetzen eines IM 151-7 CPU (6ES7 151-7Ax00-0AB0) durch ein IM 151-7 CPU (6ES7 151-7Ax10-0AB0) in der Projektierung

Wenn der Anwender an der Projektierung nichts ändert, werden beim Tausch eines IM 151-7 CPU (6ES7 151-7Ax00-0AB0) gegen ein IM 151-7 CPU (6ES7 151-7Ax10-0AB0) die funktionalen Einstellungen in der Projektierung auf Defaultwerte gesetzt.

Das bedeutet im Einzelnen:

- Das IM 151-7 CPU (6ES7 151-7Ax00-0AB0) war auf “kein DP” (d. h. stand-alone) eingestellt.
  → das IM 151-7 CPU (6ES7 151-7Ax10-0AB0) wird auf “MPI” gesetzt.
- Das IM 151-7 CPU (6ES7 151-7Ax00-0AB0) war auf “DP-Slave” eingestellt.
  → das IM 151-7 CPU (6ES7 151-7Ax10-0AB0) wird ebenfalls auf “DP-Slave” gesetzt.

Hinweis für den Tausch in HW Konfig

- IM 151-7 CPU markieren und anschließend tauschen funktioniert nicht.
- Der Tausch ist erst möglich, nachdem das Rack selektiert wurde.

Neue Funktionalitäten des IM 151-7 CPU (6ES7 151-7Ax10-0AB0)

- koexistente Schnittstelle MPI/DP (aktiv/passiv) (siehe Kapitel 7.5)
- neues Speicherkonzept (siehe Kapitel 7.4)
- Globale Datenkommunikation (siehe Kapitel 7.7)

  Mit diesem Dienst wird der zyklische Austausch von Globaldaten zwischen SIMATIC S7-CPUs (also auch IM 151-7 CPU) realisiert.
- S7-Basiskommunikation (siehe Kapitel 7.7)

  Mit diesem Dienst wird der Datenaustausch zwischen IM 151-7 CPU und kommunikationsfähigen SIMATIC-Baugruppen innerhalb einer S7-Station realisiert. Zur Unterstützung stehen die SFCs 65 bis 74 zur Verfügung.
- MMC bis 8 MByte (siehe Kapitel 7.3)
- Datenablage (siehe Kapitel 7.4.4)

  Die Daten werden mit Hilfe der SFCs 82 bis 84 auf einer MMC gespeichert und wieder in die CPU zurückgeladen.
- STEP 7-Projekt auf MMC speichern (siehe Kapitel 7.4.5)
- Neue SFBs

  Die SFBs 52 bis 54 und SFB 75 nach IEC 61784-1 werden unterstützt.
- Betriebsstundenzähler 32 Bit

  Der Zähler wird mit Hilfe des SFC 101 bedient.

Operationsliste

A

In diesem Anhang finden Sie den gesamten Operationsumfang, der Ihnen für die Programmierung des IM 151-7 CPU mit *STEP 7* zur Verfügung steht. Außerdem finden Sie zu jeder Operation ihre typische Ausführungszeit.

Eine ausführliche Beschreibung aller Operationen mit Beispielen finden Sie in den *STEP 7-Programmierhandbüchern*.

Hinweis

Bei indirekter Adressierung (Beispiele siehe Kapitel A.4) müssen Sie zu den Ausführungszeiten noch eine Zeit für das Laden der Adresse des jeweiligen Operanden addieren (siehe Kapitel A.5).

Übersicht

Anhang	Thema	Seite
A.1	Operanden und Parameterbereiche	A-2
A.2	Abkürzungen	A-3
A.3	Register	A-3
A.4	Adressierungsbeispiele	A-5
A.5	Ausführungszeiten bei indirekter Adressierung	A-7
ab A.6	Operationsliste	A-13

A.1 Operanden und Parameterbereiche

Operand	Parameterbereich	Beschreibung
A	0.0 bis 127.7	Ausgang (im PAA)
AB	0 b. 127	Ausgangsbyte (im PAA)
AW	0 b. 126	Ausgangswort (im PAA)
AD	0 b. 124	Ausgangsdoppelwort (im PAA)
B	–	bereichsübergreifend, registerindirekt adressiertes Byte
W	–	bereichsübergreifend, registerindirekt adressiertes Wort
D	–	bereichsübergreifend, registerindirekt adressiertes Doppelwort
DBX	0.0 bis 16383.7	Datenbit im Datenbaustein
DB	1 bis 511	Datenbaustein
DBB	0 bis 16383	Datenbyte im DB
DBW	0 bis 16382	Datenwort im DB
DBD	0 bis 16380	Datendoppelwort im DB
DIX	0.0 bis 16383.7	Datenbit im Instanz-DB
DI	1 bis 511	Instanz-Datenbaustein
DIB	0 bis 16383	Datenbyte im Instanz-DB
DIW	0 bis 16382	Datenwort im Instanz-DB
DID	0 bis 16380	Datendoppelwort im Instanz-DB
E	0.0 bis 127.7	Eingang (im PAE)
EB	0 b. 127	Eingangsbyte (im PAE)
EW	0 b. 126	Eingangswort (im PAE)
ED	0 b. 124	Eingangsdoppelwort (im PAE)
L	0.0 bis 509.7	Lokaldaten
LB	0 bis 509	Lokaldatenbyte
LW	0 bis 508	Lokaldatenwort
LD	0 bis 506	Lokaldaten-Doppelwort
M	0.0 bis 255.7	Merker
MB	0 b. 255	Merkerbyte
MW	0 b. 254	Merkerwort
MD	0 b. 252	Merkerdoppelwort
PAB	0 bis 2047	Peripherieausgangsbyte
PAW	0 bis 2046	Peripherieausgangswort
PAD	0 bis 2044	Peripherieausgangs-Doppelwort
PEB	0 bis 2047	Peripherieeingangsbyte
PEW	0 bis 2046	Peripherieeingangswort
PED	0 bis 2044	Peripherieeingangs-Doppelwort
T	0 bis 256	Timer (Zeiten)
Z	0 bis 256	Zähler
Parameter	–	Operand, über Parameter adressiert
B#	–	Konstante, 2 oder 4 Byte
D#	–	IEC Datumskonstante
L#	–	Ganzzahl-Konstante (32 Bit)
P#	–	Pointerkonstante
S5T#	–	S5-Zeitkonstante (16 Bit)*
T#	**	Zeitkonstante (16/32 Bit)
TOD#	–	IEC Zeitkonstante (32 Bit)
C#	–	Zählerkonstante (16/32 Bit)
2#	–	Binärkonstante (16/32 Bit)
16#	–	Hexadezimalkonstante (16/32 Bit)

* dient zum Laden der S5-Timer

** T # 1D_5M_3M_1S_2MS

A.2 Abkürzungen

Folgende Abkürzungen verwenden wir in der Operationsliste:

Abkür-zung	... steht für	Beispiel
k8	Konstante (8 Bit)	32
k16	Konstante (16 Bit)	62 531
k32	Konstante (32 Bit)	127 624
i8	Ganzzahl (8 Bit)	–155
i16	Ganzzahl (16 Bit)	+6523
i32	Ganzzahl (32 Bit)	–2 222 222
m	P#x.y (Pointer)	P#240.3
n	Binärkonstante	1001 1100
p	Hexadezimalkonstante	EA12
MARKE	symbolische Sprungadresse (max. 4 Buchstaben)	ZIEL

A.3 Register

AKKU1 und AKKU2 (32 Bit)

Die AKKUs sind Register für die Verarbeitung von Bytes, Worten oder Doppelworten. Dazu werden die Operanden in die AKKUs geladen und dort verknüpft. Das Verknüpfungsergebnis steht immer im AKKU1.

Die AKKUs sind 32 Bit breit.

Bezeichnungen:

AKKU		Bit
AKKU1	AKKU2	Bit 0 bis 31
AKKU1-L	AKKU2-L	Bit 0 bis 15
AKKU1-H	AKKU2-H	Bit 16 bis 31
AKKU1-LL	AKKU2-LL	Bit 0 bis 7
AKKU1-LH	AKKU2-LH	Bit 8 bis 15
AKKU1-HL	AKKU2-HL	Bit 16 bis 23
AKKU1-HH	AKKU2-HH	Bit 24 bis 31

Adressregister AR1 und AR2 (32 Bit)

Die Adressregister enthalten die bereichsinternen oder bereichsübergreifenden Adressen für die registerindirekt adressierenden Operationen. Die Adressregister sind 32 Bit breit.

Die bereichsinternen bzw. bereichsübergreifenden Adressen haben folgenden Aufbau:

- bereichsinterne Adresse:

  `00000000 00000bbb bbbbbbbb bbbbbxxx`

- bereichsübergreifende Adresse:

  `10000yyy` `00000bbb bbbbbbbb bbbbbxxx`

Legende:

	`b`	Byteadresse
	`x`	Bitnummer
	`y`	Bereichskennung (siehe Kap. A.4)

Statuswort (16 Bit)

Die Anzeigen werden durch die Operationen ausgewertet oder gesetzt.

Das Statuswort ist 16 Bit breit.

Bit	Belegung	Bedeutung
0	/ER	Erstabfrage *
1	VKE	Verknüpfungsergebnis
2	STA	Status *
3	OR	Oder *
4	OS	Überlauf speichernd
5	OV	Überlauf
6	A0	Ergebnisanzeige
7	A1	Ergebnisanzeige
8	BIE	Binärergebnis
9 bis 15	nicht belegt	–

* Bit kann im Anwenderprogramm mit Operation L STW nicht ausgewertet werden, da das Bit zur Programmlaufzeit nicht aktualisiert wird.

A.4 Adressierungsbeispiele

Adressierungsbeispiele	Beschreibung
Unmittelbare Adressierung	
L +27	Lade 16-Bit-Ganzzahl-Konstante “27” in AKKU1
L L#–1	Lade 32-Bit-Ganzzahl-Konstante “–1” in AKKU1
L 2#1010101010101010	Lade Binärkonstante in AKKU1
L DW#16#A0F0 BCFD	Lade Hexadezimalkonstante in AKKU1
L ’ENDE’	Lade ASCII-Zeichen in AKKU1
L T#500 ms	Lade Zeitwert in AKKU1
L P#10.0	Lade bereichsinternen Pointer in AKKU1
L P#E20.6	Lade bereichsübergreifenden Pointer in AKKU1
L –2.5	Lade Realzahl in AKKU1
L D#1997–01–20	Lade Datum
L TOD#13:20:33.125	Lade Uhrzeit
Direkte Adressierung	
U E 0.0	UND-Verknüpfung des Eingangsbits 0.0
L EB 1	Lade Eingangsbyte 1 in AKKU1
L EW 0	Lade Eingangswort 0 in AKKU1
L ED 0	Lade Eingangsdoppelwort 0 in AKKU1
Indirekte Adressierung Timer/Zähler	
SI T [LW 8]	Starte Timer; die Timer-Nr. steht im Lokaldatenwort 8
ZV Z [LW 10]	Starte Zähler; die Zähler-Nr. steht im Lokaldatenwort 10
Speicherindirekte, bereichsinterne Adressierung	
U E [LD 12] Beispiel: L P#22.2 T LD 12 U E [LD 12]	UND-Operation; die Adresse des Eingangs steht als Pointer im Lokaldaten-Doppelwort 12
U E [DBD 1]	UND-Operation; die Adresse des Eingangs steht als Pointer im Datendoppelwort 1 des DB
U A [DID 12]	UND-Operation; die Adresse des Ausgangs steht als Pointer im Datendoppelwort 12 des Instanz-DB
U A [MD 12]	UND-Operation; die Adresse des Ausgangs steht als Pointer im Merkerdoppelwort 12
Registerindirekte, bereichsinterne Adressierung	
U E [AR1,P#12.2]	UND-Operation; die Adresse des Eingangs errechnet sich zu “Pointerwert im Adressregister 1 + Pointer P#12.2”

<table>
<tr><th>Adressierungsbeispiele</th><th>Beschreibung</th></tr>
<tr><td colspan="2">Registerindirekte, bereichsübergreifende Adressierung</td></tr>
<tr><td colspan="2">Für die bereichsübergreifende, registerindirekte Adressierung muss die Adresse zusätzlich eine Bereichskennung in den Bits 24-26 enthalten. Die Adresse steht im Adressregister.

Bereichs-kennung | Codierung binär | hex. | Bereich
P | 1000 0000 | 80 | Peripheriebereich
E | 1000 0001 | 81 | Eingangsbereich
A | 1000 0010 | 82 | Ausgangsbereich
M | 1000 0011 | 83 | Merkerbereich
DB | 1000 0100 | 84 | Datenbereich
DI | 1000 0101 | 85 | Instanz-Datenbereich
L | 1000 0110 | 86 | Lokaldatenbereich
VL | 1000 0111 | 87 | Vorgänger-Lokaldatenbereich (Zugriff auf Lokaldaten des aufrufenden Bausteins siehe Seite A-5)</td></tr>
<tr><td>L B [AR1,P#8.0]</td><td>Lade Byte in AKKU1; die Adresse errechnet sich aus “Pointerwert im Adressregister 1 + Pointer P#8.0”</td></tr>
<tr><td>U [AR1,P#32.3]</td><td>UND-Operation; die Adresse des Operanden errechnet sich aus “Pointerwert im Adressregister 1 + Pointer P#32.3”</td></tr>
<tr><td colspan="2">Adressierung über Parameter</td></tr>
<tr><td>U Parameter</td><td>Der Operand wird über den Parameter adressiert</td></tr>
</table>

Pointer berechnen

Hier 2 Beispiele für das Berechnen von Pointern:

Beispiel bei Summe der Bitadressen ≦ 7:

```
LAR1 P#8.2
U E [AR1,P#10.2]
```

Ergebnis: Adressiert wird Eingang 18.4 (durch jeweilige Addition der Byte- und Bitadressen)

Beispiel bei Summe der Bitadressen > 7:

```
L MD 0      beliebig berechneter Pointer, z.B. P#10.5
LAR1
U E [AR1,P#10.7]
```

Ergebnis: Adressiert wird Eingang 21.4 (durch Addition der Byte- und Bitadressen mit Übertrag)

A.5 Ausführungszeiten bei indirekter Adressierung

Die Ausführungszeiten bei indirekter Adressierung müssen Sie berechnen. Die Berechnung erklären wir Ihnen in diesem Kapitel.

2 Teile einer Anweisung

Eine Anweisung mit indirekt adressierten Operanden besteht aus 2 Teilen:

1. Teil: Laden der Adresse des Operanden

2. Teil: Ausführen der Operation

Das bedeutet, Sie müssen auch die Ausführungszeit einer Anweisung mit indirekt adressiertem Operanden aus diesen beiden Teilen berechnen.

Ausführungszeit berechnen

Für die Gesamt-Ausführungszeit gilt:

	Ausführungszeit für das Laden der Adresse
+	Ausführungszeit der Operation
=	Gesamt-Ausführungszeit der Operation

Die in Kapitel A.6 und in den folgenden Kapiteln angegebenen Ausführungszeiten sind die Ausführungszeiten für den 2. Teil einer Anweisung, also für das eigentliche Ausführen einer Operation.

Zu dieser Zeit müssen Sie noch die Ausführungszeit für das Laden der Adresse des Operanden hinzufügen.

Laden einer Adresse

Die Ausführungszeit für das Laden der Adresse des Operanden aus den verschiedenen Bereichen finden Sie in der folgenden Tabelle.

Adresse liegt im...	Ausführungszeit in µs
Merkerbereich M	
Wort	0,4
Doppelwort	0,9
Datenbaustein DB/DI	
Wort	0,8
Doppelwort	2,0
Lokaldatenbereich L	
Wort	0,5
Doppelwort	1,2
AR1/AR2 (bereichsintern)	0,5
AR1/AR2 (bereichsübergreifend)	1,6

Adresse liegt im...	Ausführungszeit in µs
Parameter (Wort) ... für: • Zeiten • Zähler • Bausteinaufrufe	1,0
Parameter (Doppelwort) ... für: Bit, Byte, Worte und Doppelworte	2,0

A.5.1 Beispiel für speicherindirekte, bereichsinterne Adressierung

Beispiel

Beispiel: U E [DBD 12]

1. Schritt

Laden des Inhalts von DBD 12

Adresse liegt im ...	Ausführungszeit in µs
Merkerbereich M	
Wort	0,4
Doppelwort	0,9
Datenbaustein DB/DI	
Wort	0,8
Doppelwort	2,0

2. Schritt

UND-Verknüpfung des so adressierten Eingangs (Ausführungszeit steht in Kapitel A.6 und in den folgenden Kapiteln).

typische Ausführungszeit in µs	
direkte Adressierung	**indirekte Adressierung**
0,1	**1,6+**
:	**:**
Zeit für U E	

Gesamt-Ausführungszeit

Damit ergibt sich als Gesamt-Ausführungszeit:

	2,0 µs
+	1,6 µs
=	3,6 µs

A.5.2 Beispiel für registerindirekte, bereichsinterne Adressierung

Beispiel

U E [AR1, P#34.3]

1. Schritt

Laden des Inhalts von AR1 und Erhöhen um den Offset 34.3.

Adresse liegt im ...	Ausführungszeit in µs
:	:
AR1/AR2 (bereichsintern)	0,5
:	:

2. Schritt

UND-Verknüpfung des so adressierten Eingangs (Ausführungszeit steht in Kapitel A.6 und in den folgenden Kapiteln).

typische Ausführungszeit in µs	
direkte Adressierung	**indirekte Adressierung**
0,1 :	Zeit für U E **1,6+ :**

Gesamt-Ausführungszeit

Damit ergibt sich als Gesamt-Ausführungszeit:

Gesamt-Ausführungszeit:

	0,5 µs
+	1,6 µs
=	2,1 µs

A.5.3 Beispiel für registerindirekte, bereichsübergreifende Adressierung

Beispiel

U [AR1, P#23.1] ... mit P#E 1.0 im AR1

1. Schritt

Laden des Inhalts von AR1 und Erhöhen um den Offset 23.1.

Adresse liegt im ...	Ausführungszeit in µs
:	:
AR1/AR2 (bereichsübergreifend)	1,6
:	:

2. Schritt

UND-Verknüpfung des so adressierten Eingangs (Ausführungszeit steht in Kapitel A.6 und in den folgenden Kapiteln).

typische Ausführungszeit in µs	
direkte Adressierung	**indirekte Adressierung**
0,1	**1,6+**
:	Zeit für U E **:**

Gesamt-Ausführungszeit

Damit ergibt sich als Gesamt-Ausführungszeit:

Gesamt-Ausführungszeit:

	1,6 µs
+	1,6 µs
=	3,2 µs

A.5.4 Beispiel für Adressierung über Parameter

Beispiel

U Parameter ... mit E 0.5 in der Bausteinparameterliste

1. Schritt

Laden des über den Parameter adressierten E 0.5.

Adresse liegt im ...	Ausführungszeit in µs
:	:
:	:
Parameter (Doppelwort)	2,0

2. Schritt

UND-Verknüpfung des so adressierten Eingangs (Ausführungszeit steht in Kapitel A.6 und in den folgenden Kapiteln).

typische Ausführungszeit in µs	
direkte Adressierung	**indirekte Adressierung**
0,1 :	**1,6+** Zeit für U E **:**

Gesamt-Ausführungszeit

Damit ergibt sich als Gesamt-Ausführungszeit:

Gesamt-Ausführungszeit:

	2,0 µs
+	1,6 µs
=	3,6 µs

A.6 Verknüpfungsoperationen mit Bitoperanden

Abfrage des adressierten Operanden auf seinen Signalzustand und Verknüpfung des Ergebnisses mit dem VKE nach der entsprechenden Funktion.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
U		UND			
	E/A	Eingang/Ausgang	1**/2	0,1	1,6+
	M	Merker	1**/2	0,2	1,7+
	L	Lokaldatenbit	2	0,4	2,0+
	DBX/DIX	Datenbit	2	1,6	2,4+
	[AR1,m]	E/A/M/L/DBX/DIX (über AR1, AR2	2	–	+
	[AR2,m]	bereichsübergreifend oder über		–	+
	Parameter	Parameter adressiert)		–	+
UN		UND-NICHT			
	E/A	Eingang/Ausgang	2	0,2	1,7+
	M	Merker		0,3	1,8+
	L	Lokaldatenbit		0,4	2,1+
	DBX/DIX	Datenbit		1,6	2,5+
	[AR1,m]	E/A/M/L/DBX/DIX (über AR1, AR2	2	–	+
	[AR2,m]	bereichsübergreifend oder über		–	+
	Parameter	Parameter adressiert)		–	+

Statuswort für: **U, UN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	ja	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	ja	ja	ja	1

O		ODER			
	E/A	Eingang/Ausgang	1**/2	0,1	1,6+
	M	Merker	1**/2	0,2	1,7+
	L	Lokaldatenbit	2	0,4	2,0+
	DBX/DIX	Datenbit	2	1,6	2,4+
	[AR1,m]	E/A/M/L/DBX/DIX (über AR1, AR2	2	–	+
	[AR2,m]	bereichsübergreifend oder über		–	+
	Parameter	Parameter adressiert)		–	+
ON		ODER-NICHT			
	E/A	Eingang/Ausgang	1**/2	0,2	1,7+
	M	Merker	1**/2	0,3	1,8+
	L	Lokaldatenbit	2	0,4	2,1+
	DBX/DIX	Datenbit	2	1,7	2,5+
	[AR1,m]	E/A/M/L/DBX/DIX (über AR1, AR2	2	–	+
	[AR2,m]	bereichsübergreifend oder über		–	+
	Parameter	Parameter adressiert)		–	+

Statuswort für: **O, ON**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	0	ja	ja	1

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
X		EXKLUSIV-ODER			
	E/A	Ein-/Ausgang	2	0,1	1,6+
	M	Merker		0,2	1,7+
	L	Lokaldatenbit		0,4	2,0+
	DBX/DIX	Datenbit		1,6	2,4+
	[AR1,m]	E/A/M/L/DBX/DIX (über AR1, AR2	2	–	+
	[AR2,m]	bereichsübergreifend oder über Pa-		–	+
	Parameter	rameter adressiert)		–	+
XN		EXKLUSIV-ODER-NICHT			
	E/A	Eingang/Ausgang	2	0,2	1,7+
	M	Merker		0,3	1,8+
	L	Lokaldatenbit		0,4	2,1+
	DBX/DIX	Datenbit		1,7	2,5+
	[AR1,m]	E/A/M/L/DBX/DIX (über AR1, AR2	2	–	+
	[AR2,m]	bereichsübergreifend oder über Pa-		–	+
	Parameter	rameter adressiert)		–	+

Statuswort für: **X, XN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	0	ja	ja	1

* + Zeit für das Laden der Adresse des Operanden

A.7 Verknüpfungsoperationen von Klammerausdrücken

Retten der Bits BIE, VKE, OR und einer Funktionskennung (U, UN, bis) auf den Klammerstack. 7 Klammerebenen sind pro Baustein möglich.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
U(		UND-Klammer-Auf	1	0,4
UN(		UND-NICHT-Klammer-Auf	1	0,5
O(		ODER-Klammer-Auf	1	0,3
ON(		ODER-NICHT-Klammer-Auf	1	0,4
X(		EXKLUSIV-ODER-Klammer-Auf	1	0,4
XN(		EXKLUSIV-ODER-NICHT-Klammer-Auf	1	0,4

Statuswort für: **U(, UN(, O(, ON(, X(, XN(**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	ja	–	–	–	–	ja	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	0	1	–	0

)		Klammer zu, Entfernen eines Eintrags vom Klammerstack, Verknüpfen des VKE mit dem aktuellen VKE im Prozessor	1	1,0

Statuswort für: **)**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	ja	–	–	–	–	ja	1	ja	1

A.8 ODER-Verknüpfung von UND-Funktionen

Es erfolgt die ODER-Verknüpfung von UND-Funktionen nach der Regel: UND vor ODER.

Operation	**Operand**	**Bedeutung**	**Länge in Worten**	**typische Ausführungszeit in µs**
O		ODER-Verknüpfung von UND-Funktionen nach der Regel: UND vor ODER	1	0,1

Statuswort für: **O**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	ja	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	ja	1	–	ja

A.9 Verknüpfungsoperationen mit Zeiten und Zählern

Abfrage des adressierten Timers/Zählers auf seinen Signalzustand und Verknüpfen des Ergebnisses mit dem VKE nach der entsprechenden Funktion.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung *
U	T Z	UND Timer UND Zähler	1**/2	0,3 0,2	1,1+ 1,1+
	Timerpara. Zählerpara.	UND Timer/Zähler (über Parameter adressiert)	2	– –	+ +
UN	T Z	UND-NICHT Timer UND-NICHT Zähler	1**/2	0,4 0,3	1,2+ 1,2+
	Timerpara. Zählerpara.	UND-NICHT Timer/Zähler (über Parameter adressiert)	2	– –	+ +

Statuswort für: **U, UN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	ja	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	ja	ja	ja	1

O	T Z	ODER Timer ODER Zähler	1**/2	0,3 0,2	1,1+ 1,0+
	Timerpara. Zählerpara.	ODER Timer/Zähler (über Parameter adressiert)	2	– –	+ +
ON	T Z	ODER-NICHT Timer ODER-NICHT Zähler	1**/2	0,4 0,3	1,2+ 1,1+
	Timerpara. Zählerpara.	ODER-NICHT Timer/Zähler (über Parameter adressiert)	2	– –	+ +
X	T Z	EXKLUSIV-ODER Timer EXKLUSIV-ODER Zähler	2	0,3 0,2	1,1+ 1,1+
	Timerpara. Zählerpara	EXKLUSIV-ODER Timer/Zähler (über Parameter adressiert)	2	– –	+ +
XN	T Z	EXKLUSIV-ODER-NICHT Timer EXKLUSIV-ODER-NICHT Zähler	2	0,4 0,3	1,2+ 1,2+
	Timerpara. Zählerpara.	EXKLUSIV-ODER-NICHT Timer/Zähler (über Parameter adressiert)	2	– –	+ +

Statuswort für: **O, ON, X, XN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	0	ja	ja	1

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung

A.10 Verknüpfungsoperationen mit dem Inhalt von AKKU1

Verknüpfung des Inhalts von AKKU1 bzw. AKKU1-L mit einem Wort bzw. einem Doppelwort nach der entsprechenden Funktion. Das Wort bzw. Doppelwort steht entweder als Konstante in der Operation oder im AKKU2. Das Ergebnis steht im AKKU1 bzw. AKKU1-L.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
UW		UND AKKU2-L	1	0,3
	k16	UND 16-Bit-Konstante	2	0,3
OW		ODER AKKU2-L	1	0,3
	k16	ODER 16-Bit-Konstante	2	0,3
XOW		EXKLUSIV-ODER AKKU2-L	1	0,3
	k16	EXKLUSIV-ODER 16-Bit-Konstante	2	0,3
UD		UND AKKU2	1	0,9
	k32	UND 32-Bit-Konstante	3	1,0
OD		ODER AKKU2	1	0,9
	k32	ODER 32-Bit-Konstante	3	1,0
XOD		EXKLUSIV-ODER AKKU2	1	0,9
	k32	EXKLUSIV-ODER 32-Bit-Konstante	3	1,0

Statuswort für: **UW, OW, XOW, UD, OD, XOD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	0	0	–	–	–	–	–

A.11 Verknüpfungsoperationen mit Anzeigenbits

Abfrage der angegebenen Bedingungen auf ihren Signalzustand und Verknüpfen des Ergebnisses mit dem VKE nach der entsprechenden Funktion.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
U, O, X	==0	UND Ergebnis=0 (A1=0) and (A0=0)	1	0,2
	>0	UND Ergebnis>0 (A1=1) and (A0=0)	1	0,3
	<0	UND Ergebnis<0 (A1=0) and (A0=1)	1	0,3
	<>0	UND Ergebnis ≠ 0 ((A1=0) and (A0=1) or (A1=1) and (A0=0))	1	0,2
	<=0	UND Ergebnis<=0 ((A1=0) and (A0=1) or (A1=0) and (A0=0))	1	0,2
	>=0	UND Ergebnis>=0 ((A1=1) and (A0=0) or (A1=0) and (A0=0))	1	0,2
	UO	UND unordered/unzulässig (A1=1) and (A0=1)	1	0,2
	OS	UND OS=1	1	0,1
	BIE	UND BIE=1	1	0,1
	OV	UND OV=1	1	0,1

Statuswort für: **U , O, X**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	ja	ja	ja	ja	ja	ja	–	ja	ja
Operation beeinflusst:	–	–	–	–-	–	ja	ja	ja	1

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
UN, ON, XN	==0	UND-NICHT Ergebnis=0 (A1=0) and (A0=0)	1	0,2
	>0	UND-NICHT Ergebnis>0 (A1=1) and (A0=0)	1	0,3
	<0	UND-NICHT Ergebnis<0 (A1=0) and (A0=1)	1	0,3
	<>0	UND-NICHT Ergebnis ≠ 0 ((A1=0) and (A0=1) or (A1=1) and (A0=0))	1	0,3
	<=0	UND-NICHT Ergebnis<=0 ((A1=0) and (A0=1) or (A1=0) and (A0=0))	1	0,1
	>=0	UND-NICHT Ergebnis>=0 ((A1=1) and (A0=0) or (A1=0) and (A0=0))	1	0,1
	UO	UND-NICHT unordered/unzulässig (A1=1) and (A0=1)	1	0,3
	OS	UND-NICHT OS=1	1	0,2
	OV	UND-NICHT OV=1	1	0,2
	BIE	UND-NICHT BIE=1	1	0,2

Statuswort für: **UN, ON, XN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	ja	ja	ja	ja	ja	ja	–	ja	ja
Operation beeinflusst:	–	–	–	–	–	ja	ja	ja	1

A.12 Flankenoperationen

Erkennen eines Flankenwechsels. Der aktuelle Signalzustand im VKE wird vergleichen mit dem Signalzustand im Operanden, dem “Flankenmerker”.
FP erkennt einen Flankenwechsel von “0” nach “1”.
FN erkennt einen Flankenwechsel von “1” nach “0”.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
FP	E/A M L DBX/DIX [AR1,m] [AR2,m] Parameter	Anzeigen der steigenden Flanke im VKE. Flankenhilfsmerker ist das in der Operation adressierte Bit.	2	0,3 0,6 0,7 1,9 – – –	1,8+ 1,9+ 2,1+ 2,7+ + + +
FN	E/A M L DBX/DIX [AR1,m] [AR2,m] Parameter	Anzeigen der fallenden Flanke im VKE. Flankenhilfsmerker ist das in der Operation adressierte Bit.	2	0,3 0,6 0,7 1,9 – – –	1,9+ 2,0+ 2,2+ 2,8+ + + +

Statuswort für: **FP, FN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	–	0	ja	ja	1

* + Zeit für das Laden der Adresse des Operanden

A.13 Setzen/Rücksetzen von Bitoperanden

Zuweisen des Wertes “1” oder “0” bzw. des VKE an den adressierten Operanden. Die Operationen können vom MCR abhängig sein.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
S	E/A	Setze Eingang/Ausgang auf “1”	1**/2	0,1	1,7+
		(MCR-abhängig)		0,2	1,8+
	M	Setze Merker auf “1”	1**/2	0,3	1,8+
		(MCR-abhängig)		1,0	2,0+
	L	Setze Lokaldatenbit auf “1”	2	0,5	2,0+
		(MCR-abhängig)		1,0	2,1+
	DBX/DIX	Setze Datenbit auf “1”	2	1,7	2,6+
		(MCR-abhängig)		1,8	2,7+
	[AR1,m] [AR2,m] Parameter	Setze E/A/M/L/DBX/DIX (MCR-abhängig) auf ”1”. (über AR1, AR2 bereichsübergreifend oder über Parameter adressiert)	2	– – –	+ + +
R	E/A	Rücksetze Eingang/Ausgang auf “0”	1**/2	0,1	1,7+
		(MCR-abhängig)		0,2	1,8+
	M	Rücksetze Merker auf “0”	1**/2	0,3	1,8+
		(MCR-abhängig)		1,0	1,9+
	L	Rücksetze Lokaldatenbit auf “0”	2	0,5	2,1+
		(MCR-abhängig)		1,1	2,1+
	DBX/DIX	Rücksetze Datenbit auf “0”	2	1,8	2,6+
		(MCR-abhängig)		1,8	2,7+
	[AR1,m] [AR2,m] Parameter	Setze E/A/M/L/DBX/DIX (MCR–abhängig) auf “0”. (über AR1, AR2 bereichsübergreifend oder über Parameter adressiert)	2	– – –	+ + +
=	E/A	Zuweisen des VKE an Eingang/Ausgang	1**/2	0,1	1,7+
		(MCR-abhängig)		0,2	1,8+
	M	Zuweisen des VKE an Merker	1**/2	0,4	1,8+
		(MCR-abhängig)		2,0	2,0+
	L	Zuweisen des VKE an Lokaldatenbit	2	0,4	2,0+
		(MCR-abhängig)		1,0	2,2+
	DBX/DIX	Zuweisen des VKE an Datenbit	2	1,8	2,6+
		(MCR-abhängig)		1,9	2,7+
	[AR1,m] [AR2,m] Parameter	Zuweisen des VKE an E/A/M/L/DBX/DIX (über AR1, AR2 bereichsübergreifend oder über Parameter adressiert)	2	–	+

Statuswort für: **S, R, =**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	–	0	ja	–	0

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung

A.14 VKE direkt beeinflussende Operationen

Die folgenden Operationen bearbeiten direkt das VKE.

Operation	**Operand**	**Bedeutung**	**Länge in Worten**	**typische Ausführungszeit in µs**
CLR		Setze VKE auf “0”	2	0,1

Statuswort für: **CLR**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	0	0	0	0

SET		Setze VKE auf “1”	2	0,1

Statuswort für: **SET**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	0	1	1	0

NOT		Negiere das VKE	2	0,1

Statuswort für: **NOT**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	ja	–	ja	–
Operation beeinflusst:	–	–	–	–	–	–	1	ja	–

SAVE		Rette das VKE in das BIE-Bit	1	0,1

Statuswort für: **SAVE**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	ja	–	–	–	–	–	–	–	–

A.15 Zeitoperationen

Starten bzw. Rücksetzen eines Timers (direkt adressiert oder über Parameter adressiert). Die Zeitdauer muss im AKKU1-L stehen.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
SI	T	Starte Zeit als Impuls bei Flankenwechsel von “0” nach “1”	4**/6	2,3	2,9+
	Timer-para.		2	–	+
SV	T	Starte Zeit als verlängerten Impuls bei Flankenwechsel von “0” nach “1”	4**/6	1,1	1,2+
	Timer-para.		2	–	+
SE	T	Starte Zeit als Einschaltverzögerung bei Flankenwechsel von “0” nach “1”	4**/6	2,4	3,0+
	Timer-para.		2	–	+
SS	T	Starte Zeit als speichernde Einschaltverzögerung bei Flankenwechsel von “0” nach “1”	4**/6	2,4	3,0+
	Timer-para.		2	–	+
SA	T	Starte Zeit als Ausschaltverzögerung bei Flankenwechsel von “1” nach “0”	4**/6	2,6	3,2+
	Timer-para.		2	–	+
FR	T	Freigabe eines Timers für das erneute Starten bei Flankenwechsel von “0” nach “1” (Löschen des Flankenmerkers für das Starten der Zeit)	4**/6	1,1	1,5+
	Timer-para.		2	–	+
R	T	Rücksetzen einer Zeit	4**/6	1,1	1,5+
	Timer-para.		2	–	+

Statuswort für: **SI, SV, SE, SS, SA, FR, R**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	–	0	–	–	0

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung des Operanden

A.16 Zähloperationen

Der Zählwert steht im AKKU1-L bzw. in der als Parameter übergebenen Adresse.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung *
S	Z	Vorbelegen eines Zählers bei Flankenwechsel von “0” nach “1”	4**/6	1,8	2,4+
	Zählerpara.		2	–	+
R	Z	Rücksetzen des Zählers auf “0” bei Flankenwechsel von “0” nach “1”	4**/6	0,7	1,1+
	Zählerpara.		2	–	+
ZV	Z	Zähle um 1 vorwärts bei Flankenwechsel von “0” nach “1”	4**/6	1,1	1,6+
	Zählerpara.		2	–	+
ZR	Z	Zähle um 1 rückwärts bei Flankenwechsel von “0” nach “1”	4**/6	1,1	1,5+
	Zählerpara.		2	–	+
FR	Z	Freigabe eines Zählers bei Flankenwechsel von “0” nach “1” (Löschen des Flankenmerkers für Vorwärts- und Rückwärtszählen eines Zählers)	2	0,9	1,4+
	Zählerpara.		2	–	+

Statuswort für: **S, R, ZV, ZR, FR**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	–	0	–	–	0

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung des Operanden

A.17 Ladeoperationen

Laden der Operanden in AKKU1. Zuvor wird der alte Inhalt von AKKU1 in AKKU2 gerettet.
Das Statuswort wird nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs direkte Adressierung	typische Ausführungszeit in µs indirekte Adressierung*
L		Lade ...			
	EB	Eingangsbyte	1**/2	0,2	1,4+
	AB	Ausgangsbyte	1**/2	0,2	1,4+
	PEB	Peripherie-Eingangsbyte	1**/2	< 125	< 127
	MB	Merkerbyte	1**/2	0,3	1,4+
	LB	Lokaldatenbyte	2	0,5	1,7+
	DBB	Datenbyte	2	1,5	2,5+
	DIB	Instanz-Datenbyte	2	1,5	2,5+
		... in AKKU1			
	Parameter	Lade EB/AB/PEB/MB/LB/DBB/DIB in AKKU1 (über Parameter adressiert)	2	–	+
		Lade ...			
	EW	Eingangswort	1**/2	0,3	1,6+
	AW	Ausgangswort	1**/2	0,3	1,6+
	PEW	Peripherie-Eingangswort	1**/2	< 135	< 137
	MW	Merkerwort	1**/2	0,6	1,7+
	LW	Lokaldatenwort	2	0,6	2,0+
	DBW	Datenwort	1**/2	1,9	3,0+
	DIW	Instanz-Datenwort	1**/2	1,9	3,0+
		... in AKKU1-L			
	Parameter	Lade EW/AW/PEW/MW/LW/DBW/DIW in AKKU1 (über Parameter adressiert)	2	–	+
		Lade ...			
	ED	Eingangsdoppelwort	1**/2	0,4	1,6+
	AD	Ausgangsdoppelwort	1**/2	0,4	1,6+
	PED	Peripherie-Eingangsdoppelwort	2	< 145	< 147
	MD	Merkerdoppelwort	1**/2	0,8	2,0+
	LD	Lokaldatendoppelwort	2	0,9	2,3+
		Datendoppelwort	2	2,5	3,7+
	DBD	Instanz-Datendoppelwort	2	2,5	3,7+
	DID	... in AKKU1			
	Parameter	Lade ED/AD/PED/MD/LD/DBD/DID in AKKU1 (über Parameter adressiert)	2	–	+

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung des Operanden

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
L		Lade bereichsübergreifend adressiert....			
	B[AR1,m]	Byte			40,1+
	B[AR2,m]		2	–	40,1+
	W[AR1,m]	Wort			45,6+
	W[AR2,m]		2	–	45,6+
	D[AR1,m]	Doppelwort			57,4+
	D[AR2,m]		2	–	57,4+
		... in AKKU1			
		Lade ...			
	k8	8-Bit-Konstante in AKKU1-LL	1	0,2	–
	k16	16-Bit-Konstante in AKKU1-L	2	0,2	–
	L#k32	32-Bit-Konstante in AKKU1	3	0,3	–
	Parameter	Lade Konstante in AKKU1 (über Parameter adressiert)	2	–	+
	2#n	Lade 16-Bit-Binärkonstante in AKKU1-L	2	0,2	–
		Lade 32-Bit-Binärkonstante in AKKU1	3	0,3	–
	16#p	Lade 16-Bit-Hexadezimalkonstante in AKKU1-L	2	0,2	–
		Lade 32-Bit-Hexadezimalkonstante in AKKU1	3	0,3	–
				typische Ausführungszeit in µs	
	'xx'	Lade 2 Zeichen	2	0,2	
	'xxxx'	Lade 4 Zeichen	3	0,3	
	D#Datum	Lade IEC-Datum (BCD-codiert)	3	0,3	
	S5T# Zeitwert	Lade S5-Zeitkonstante (16 Bit)	2	0,3	
	TOD# Zeitwert	Lade 32-Bit-Zeitkonstante (IEC-Tageszeit)	3	0,3	
	T# Zeitwert	Lade 16-Bit-Zeitkonstante	2	0,2	
		Lade 32-Bit-Zählerkonstante	3	0,3	
	C# Zählwert	Lade 16-Bit-Zählerkonstante	2	0,2	
		Lade 32-Bit-Zählerkonstante	3	0,3	
	P# Bitpointer	Lade Bitpointer	3	0,3	
	L# Integer	Lade 32-Bit-Ganzzahl-Konstante	3	0,3	
	q	Lade Realzahl	3	0,3	

* +Zeit für das Laden der Adresse des Operanden

A.18 Ladeoperationen für Timer und Zähler

Laden eines Zeitwertes oder Zählwertes in AKKU1. Zuvor wird der Inhalt von AKKU1 in AKKU2 gerettet.
Die Anzeigen werden nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
L	T	Lade Zeitwert	1**/2	0,8	1,1+
	Timer-para.	Lade Zeitwert (über Parameter adressiert)	2	–	+
	Z	Lade Zählwert	1**/2	0,8	1,2+
	Zähler-para.	Lade Zählwert (über Parameter adressiert)	2	–	+
LC	T	Lade Zeitwert BCD-codiert	1**/2	2,2	2,5+
	Timer-para.	Lade Zeitwert BCD-codiert (über Parameter adressiert)	2	–	+
	Z	Lade Zählwert BCD-codiert	1**/2	2,3	2,9+
	Zähler-para.	Lade Zählwert (über Parameter adressiert)	2	–	+

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung des Operanden

A.19 Transferoperationen

Transferieren des Inhalts von AKKU1 in den adressierten Operanden. Das Statuswort wird nicht beeinflusst. Beachten Sie, dass einige Transferoperationen vom MCR abhängen.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs	
				direkte Adressierung	indirekte Adressierung*
T		Transferiere Inhalt von AKKU1-LL zum ...			
	EB	Eingangsbyte	1**/2	0,1	1,3+
		(MCR-abhängig)		1,2	1,5+
	AB	Ausgangsbyte	1**/2	0,1	1,3+
		(MCR-abhängig)		1,2	1,5+
	PAB	Peripherie-Ausgangsbyte	1***/2	< 125	< 127
		(MCR-abhängig)		< 126	< 128
	MB	Merkerbyte	1**/2	0,1	1,3+
		(MCR-abhängig)		0,6	1,5+
	LB	Lokaldatenbyte	2	0,2	1,7+
		(MCR-abhängig)	2	0,6	1,5+
	DBB	Datenbyte		1,3	2,2+
		(MCR-abhängig)	2	1,4	2,4+
	DIB	Instanz-Datenbyte		1,3	2,2+
		(MCR-abhängig)		1,4	2,4+
	B[AR1,µ] B[AR2,m] Parameter	Transferiere Inhalt von AKKU1–LL zum EB/AB/PAB/MB/LB/DBB/DIB (über AR1, AR2 bereichsübergreifend oder über Parameter adressiert)	2	–	+

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung des Operanden
*** direkte Adressierung bei PAB 0 bis 255

Operation	Operand	Bedeutung	Länge in Worten	direkte Adressierung	indirekte Adressierung*
T		Transferiere Inhalt von AKKU1-L zum ...			
	EW	Eingangswort	1**/2	0,2	1,4+
		(MCR-abhängig)		0,6	1,5+
	AW	Ausgangswort	1**/2	0,2	1,4+
		(MCR-abhängig)		0,6	1,5+
	PAW	Peripherie-Ausgangswort	1***/2	< 135	< 137
		(MCR-abhängig)		< 136	< 138
	MW	Merkerwort	1**/2	0,3	1,7+
		(MCR-abhängig)		0,7	1,9+
	LW	Lokaldatenwort	2	0,3	2,0+
		(MCR-abhängig)		0,8	1,8+
	DBW	Datenwort	2	1,5	2,6+
		(MCR-abhängig)		1,6	2,8+
	DIW	Instanz-Datenwort	2	1,5	2,6+
		(MCR-abhängig)		1,6	2,8+
	W[AR1,m] W[AR2,m] Parameter	Transferiere Inhalt von AKKU1-L zum EW/ AW/PAW/MW/LW/DBW/DIW (über AR1, AR2 bereichsübergreifend oder über Parameter adressiert)	2	–	+
		Transferiere Inhalt von AKKU1 zum ...			
	ED	Eingangsdoppelwort	1**/2	0,3	1,5+
		(MCR-abhängig)		0,7	1,7+
	AD	Ausgangsdoppelwort	1**/2	0,3	1,5+
		(MCR-abhängig)		0,7	1,7+
	PAD	Peripherie-Ausgangsdoppelwort	1**/2	< 150	< 152
		(MCR-abhängig)		< 151	< 153
	MD	Merkerdoppelwort	1**/2	0,6	2,0+
		(MCR-abhängig)		1,0	2,3+
	LD	Lokaldatendoppelwort	2	0,6	2,4+
		(MCR-abhängig)		1,1	2,1+
	DBD	Datendoppelwort	2	1,9	3,0+
		(MCR-abhängig)		2,0	3,3+
	DID	Instanz-Datendoppelwort	2	1,9	3,0+
		(MCR-abhängig)		2,0	3,3+
	D[AR1,m] D[AR2,m] Parameter	Transferiere Inhalt von AKKU1 zum ED/ AD/PAD/MD/LD/DBD/DID (über AR1, AR2 bereichsübergreifend oder über Parameter adressiert)	2	–	+

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung des Operanden
*** direkte Adressierung bei PAW 0 bis 254

A.20 Lade- und Transferoperationen für Adressregister

Laden eines Doppelwortes aus einem Speicher oder einem Register in AR1 oder AR2 bzw. Transferieren eines Doppelwortes aus AR1 oder AR2 in einen Speicher oder ein Register.
Das Statuswort wird nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
LAR1		Lade Inhalt aus ...		
	–	AKKU1	1	0,1
	AR2	Adressregister 2	1	0,1
	DBD	Datendoppelwort	2	2,4
	DID	Instanz-Datendoppelwort	2	2,5
	m	32-Bit-Konstante als Pointer	3	0,2
	LD	Lokaldatendoppelwort	2	0,9
	MD	Merkerdoppelwort	2	0,8
		... in AR1		
LAR2		Lade Inhalt aus ...		
	–	AKKU1	1	0,1
	DBD	Datendoppelwort	2	0,1
	DID	Instanz-Datendoppelwort	2	2,4
	m	32-Bit-Konstante als Pointer	3	2,5
	LD	Lokaldatendoppelwort	2	0,2
	MD	Merkerdoppelwort	2	0,9
		... in AR2		0,8
TAR1		Transferiere Inhalt aus AR1 in ...		
	–	AKKU1	1	0,2
	AR2	Adressregister 2	1	0,1
	DBD	Datendoppelwort	2	1,9
	DID	Instanz-Datendoppelwort	2	1,9
	LD	Lokaldatendoppelwort	2	0,6
	MD	Merkerdoppelwort	2	0,6
TAR2		Transferiere Inhalt aus AR2 in ...		
	–	AKKU1	1	0,2
	DBD	Datendoppelwort	2	0,1
	DID	Instanz-Datendoppelwort	2	1,9
	LD	Lokaldatendoppelwort	2	1,9
	MD	Merkerdoppelwort	2	0,6
TAR		Tausche die Inhalte von AR1 und AR2	1	0,3

A.21 Lade- und Transferoperationen für das Statuswort

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
L	STW	Lade Statuswort* in AKKU1		0,6

Statuswort für: **L STW**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	ja	ja	ja	ja	ja	0	0	ja	0
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

T	STW	Transferiere AKKU1 (Bits 0 bis 8) in das Statuswort*		0,6

Statuswort für: **T STW**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	ja	ja	ja	ja	ja	0	0	ja	0

* Aufbau des Statuswortes

A.22 Ladeoperationen für DB-Nummer und DB-Länge

Laden der Nummer/Länge eines Datenbausteins in AKKU1. Der alte Inhalt von AKKU1 wird in AKKU2 gerettet.
Die Anzeigen werden nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
L	DBNO	Lade Nummer des Datenbausteins	1	1,3
L	DINO	Lade Nummer des Instanz-Datenbausteins	1	1,3
L	DBLG	Lade Länge des Datenbausteins in Byte	1	0,3
L	DILG	Lade Länge des Instanz-Datenbausteins in Byte	1	0,3

A.23 Festpunktarithmetik (16-Bit)

Arithmetische Operationen zweier 16-Bit-Zahlen. Das Ergebnis steht im AKKU1–L.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
+I	–	Addiere 2 Ganzzahlen (16 Bit) (AKKU1-L)=(AKKU1-L)+ (AKKU2-L)	1	0,5
–I	–	Subtrahiere 2 Ganzzahlen (16 Bit) (AKKU1-L)=(AKKU2-L)– (AKKU1-L)	1	0,7
*I	–	Multipliziere 2 Ganzzahlen (16 Bit) (AKKU1-L)=(AKKU2-L)* (AKKU1-L)	1	1,0
/I	–	Dividiere 2 Ganzzahlen (16 Bit) (AKKU1-L)=(AKKU2-L): (AKKU1-L) Im AKKU1-H steht der Rest der Division.	1	1,2

Statuswort für: **+I, –I,*I, /I**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	ja	ja	–	–	–	–

A.24 Festpunktarithmetik (32 Bit)

Arithmetische Operationen zweier 32-Bit-Zahlen. Das Ergebnis steht im AKKU1.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
+D	–	Addiere 2 Ganzzzahlen (32 Bit) (AKKU1)=(AKKU2)+(AKKU1)	1	0,7
–D	–	Subtrahiere 2 Ganzzahlen (32 Bit) (AKKU1)=(AKKU2)–(AKKU1)	1	1,0
*D	–	Multipliziere 2 Ganzzahlen (32 Bit) (AKKU1)=(AKKU2)*(AKKU1)	1	3,5
/D	–	Dividiere 2 Ganzzahlen (32 Bit) (AKKU1)=(AKKU2):(AKKU1)	1	2,7
MOD	–	Dividiere 2 Ganzzahlen (32 Bit) und lade den Rest der Division in AKKU1: (AKKU1)=Rest von [(AKKU2):(AKKU1)]	1	1,9

Statuswort für: **+D, –D,*D, /D, MOD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	ja	ja	–	–	–	–

A.25 Gleitpunktarithmetik (32 Bit)

Das Ergebnis der arithmetischen Operationen steht im AKKU1. Die Ausführungszeit der Operation hängt ab vom Wert, der berechnet werden soll.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
+R	–	Addiere 2 Realzahlen (32 Bit) (AKKU1)=(AKKU2)+(AKKU1)	1	< 15
–R	–	Subtrahiere 2 Realzahlen (32 Bit) (AKKU1)=(AKKU2)–(AKKU1)	1	< 15
*R	–	Multipliziere 2 Realzahlen (32 Bit) (AKKU1)=(AKKU2)*(AKKU1)	1	< 12
/R	–	Dividiere 2 Realzahlen (32 Bit) (AKKU1)=(AKKU2):(AKKU1)	1	< 15

Statuswort für: **+R, –R,*R, /R**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	ja	ja	–	–	–	–

NEGR	–	Negiere Realzahl im AKKU1	1	0,3
ABS	–	Bilde Betrag der Realzahl im AKKU1	1	0,3

Statuswort für: **NEGR, ABS**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

A.26 Addition von Konstanten

Addition von Integer-Konstanten zum AKKU1. Die Anzeigen werden nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
+	i8	Addiere eine 8-Bit-Ganzzahl-Konstante	1	0,1
+	i16	Addiere eine 16-Bit-Ganzzahl-Konstante	2	0,1
+	i32	Addiere eine 32-Bit-Ganzzahl-Konstante	3	0,2

A.27 Addition über Adressregister

Addition einer Ganzzahl (16 Bit) zum Inhalt des Adressregisters. Der Wert steht in der Operation oder im AKKU 1-L. Die Anzeigen werden nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
+AR1		Addiere Inhalt von AKKU1-L zum AR1	1	0,1
+AR1	m	Addiere Pointer-Konstante zum AR1	2	0,2
+AR2		Addiere Inhalt von AKKU1-L zum AR2	1	0,1
+AR2	m	Addiere Pointer-Konstante zum AR2	2	0,2

A.28 Vergleichsoperationen mit Ganzzahlen (16 Bit)

Vergleich der Ganzzahl (16 Bit) in AKKU1-L und AKKU2-L. VKE=1, wenn Bedingung erfüllt.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
==I		AKKU2-L=AKKU1-L	1	0,6
<>I		AKKU2-L ≠ AKKU1-L	1	0,7
<I		AKKU2-L<AKKU1-L	1	0,7
<=I		AKKU2-L<=AKKU1-L	1	0,6
>I		AKKU2-L>AKKU1-L	1	0,5
>=I		AKKU2-L>=AKKU1-L	1	0,6

Statuswort für: **==I, <>I, <I, <=I, >I, >=I**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	0	–	0	ja	ja	1

A.29 Vergleichsoperationen mit Ganzzahl (32 Bit)

Vergleich der Ganzzahl (32 Bit) in AKKU1 und AKKU2. VKE=1, wenn Bedingung erfüllt.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
==D		AKKU2=AKKU1	1	0,6
<>D		AKKU2 ≠ AKKU1	1	0,6
<D		AKKU2<AKKU1	1	0,6
<=D		AKKU2<=AKKU1	1	0,6
>D		AKKU2>AKKU1	1	0,6
>=D		AKKU2>=AKKU1	1	0,6

Statuswort für: **==D,< >D, <D, <=D, >D, >=D**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	0	–	0	ja	ja	1

A.30 Vergleichsoperationen (32-Bit-Realzahlen)

Vergleich der 32-Bit-Realzahlen in AKKU1 und AKKU2. VKE=1, wenn Bedingung erfüllt. Die Ausführungszeit der Operation hängt ab vom Wert, der verglichen werden soll.

Operation	**Operand**	**Bedeutung**	**Länge in Worten**	**typische Ausführungszeit in µs**
==R		AKKU2=AKKU1	1	8,4
<>R		AKKU2 ≠ AKKU1	1	8,4
<R		AKKU2<AKKU1	1	8,5
<=R		AKKU2<=AKKU1	1	8,3
>R		AKKU2>AKKU1	1	8,4
>=R		AKKU2>=AKKU1	1	8,4

Statuswort für: **==R,< >R, <R, <=R, >R, >=R**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	ja	ja	0	ja	ja	1

A.31 Schiebeoperationen

Schiebe Inhalt von AKKU1 oder AKKU1-L um die angegebene Anzahl von Stellen nach links/rechts. Ist kein Operand angegeben, schiebe Anzahl in AKKU2-LL. Freiwerdende Stellen werden mit Nullen bzw. mit dem Vorzeichen aufgefüllt. Zuletzt geschobenes Bit steht im Anzeigenbit A1.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
SLW		Schiebe Inhalt von AKKU1-L nach links	1	0,9
SLW	0 bis 15			0,3
SLD		Schiebe Inhalt von AKKU1 nach links	1	1,1
SLD	0 bis 32			1,2
SRW		Schiebe Inhalt von AKKU1-L nach rechts	1	0,9
SRW	0 bis 15			0,3
SRD		Schiebe Inhalt von AKKU1 nach rechts	1	1,1
SRD	0 bis 32			1,2
SSI		Schiebe Inhalt von AKKU1-L mit Vorzeichen nach rechts	1	0,8
SSI	0 bis 15			0,3
SSD		Schiebe Inhalt von AKKU1 mit Vorzeichen nach rechts	1	1,1
SSD	0 bis 32			1,2

Statuswort für: **SLW, SLD, SRW, SRD, SSI, SSD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	ja	–	–	–	–	–

A.32 Rotieroperationen

Rotiere Inhalt von AKKU1 um die angegebene Anzahl von Stellen nach links/ rechts. Ist kein Operand angegeben, rotiere Anzahl in AKKU2-LL.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
RLD		Rotiere Inhalt von AKKU1 nach links	1	1,0
RLD	0 bis 32			1,5
RRD		Rotiere Inhalt von AKKU1 nach rechts	1	1,0
RRD	0 bis 32			1,1

Statuswort für: **RLD, RRD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	ja	–	–	–	–	–

RLDA		Rotiere Inhalt von AKKU1 um eine Bitposition nach links über Anzeigebit A1		0,7
RRDA		Rotiere Inhalt von AKKU1 um eine Bitposition nach rechts über Anzeigebit A1		0,8

Statuswort für: **RLDA, RRDA**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	0	0	–	–	–	–	–

A.33 AKKU-Transferoperationen, Inkrementieren, Dekrementieren

Statuswort

Das Statuswort wird nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
TAW		Umkehr der Reihenfolge der Bytes im AKKU1-L. Aus LL, LH wird LH, LL.	1	0,1
TAD		Umkehr der Reihenfolge der Bytes in AKKU1. Aus LL, LH, HL, HH wird HH, HL, LH, LL.	1	0,2
TAK		Tausche Inhalte von AKKU1 und AKKU2	1	0,3
PUSH		Der Inhalt von AKKU1 wird in AKKU2 übertragen	1	0,1
POP		Der Inhalt von AKKU2 wird in AKKU1 übertragen	1	0,1
INC	0 bis 255	Inkrementiere AKKU1-LL	1	0,1
DEC	0 bis 255	Dekrementiere AKKU1-LL	1	0,1

A.34 Bildoperation, Nulloperation

Statuswort

Das Statuswort wird nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
BLD	0 bis 255	Bildaufbau-Operation; wird von der CPU wie eine Nulloperation behandelt. Die Anzeigen werden nicht beeinflusst.	1	0,1
NOP	0 1	Nulloperation; Anzeigen werden nicht beeinflusst.	1	0,1 0,1

A.35 Datentyp-Umwandlungsoperationen

Ergebnis

Die Ergebnisse der Wandlung stehen im AKKU1.

Bei der Wandlung von Realzahlen ist die Ausführungszeit abhängig vom Wert.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
BTI	–	Konvertiere AKKU1 von BCD nach Ganzzahl (16 Bit) (**B**CD **T**o **I**nt)	1	1,9
BTD	–	Konvertiere AKKU1 von BCD nach Ganzzahl (32 Bit) (**B**CD **T**o **D**oubleint)	1	4,1
DTR	–	Konvertiere AKKU1 von Ganzzahl (32 Bit) nach Real (32 Bit) (**D**oubleint **T**o **R**eal)	1	3,1
ITD	–	Konvertiere AKKU1 von Ganzzahl (16 Bit) nach Ganzzahl (32 Bit) (**I**nt **T**o **D**oubleint)	1	0,1

Statuswort für: **BTI, BTD, DTR, ITD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
ITB	–	Konvertiere AKKU1 von Ganzzahl (16 Bit) nach BCD (**I**nt **T**o **B**CD)	1	2,1
DTB	–	Konvertiere AKKU1 von Ganzzahl (32 Bit) nach BCD (**D**oubleint **T**o **B**CD)	1	4,8

Statuswort für: **ITB, DTB**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	ja	ja	–	–	–	–

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
RND	–	Wandle Realzahl in 32-Bit-Ganzzahl um.	1	3,4
RND–	–	Wandle Realzahl in 32-Bit-Ganzzahl um. Es wird gerundet zur nächsten ganzen Zahl.	1	3,5
RND+	–	Wandle Realzahl in 32-Bit-Integerzahl um. Es wird gerundet zur nächsten ganzen Zahl.	1	3,5
TRUNC	–	Wandle Realzahl in 32-Bit-Integerzahl um. Es werden die Nachkommazahlen abgeschnitten.	1	3,3

Statuswort für: **RND, RND–, RND+, TRUNC**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	ja	ja	–	–	–	–

A.36 Komplementbildung

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
INVI		Bilde 1er-Komplement von AKKU1-L	1	0,1
INVD		Bilde 1er-Komplement von AKKU1	1	0,1

Statuswort für: **INVI, INVD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

NEGI		Bilde 2er-Komplement von AKKU1-L (16-Bit-Ganzzahl)	1	0,6
NEGD		Bilde 2er-Komplement von AKKU1 (32-Bit-Ganzzahl)	1	1,7

Statuswort für: **NEGI, NEGD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	ja	ja	ja	ja	–	–	–	–

A.37 Baustein-Aufrufoperationen

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs: direkte Adressierung	typische Ausführungszeit in µs: indirekte Adressierung*
CALL	FC	Unbedingter Aufruf einer Function	1	8,2	-
	SFC	Unbedingter Aufruf einer Function des Betriebssystems	2	Ausführungszeiten siehe Anhang B	
UC	FC	Unbedingter Aufruf von Bausteinen ohne Parameter	1**/2	6,0	6,4+

Statuswort für: **CALL, UC**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	0	0	1	–	0

CC	FC	Bedingter Aufruf von Bausteinen ohne Parameter	1**/2	6,2	6,6+

Statuswort für: **CC**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	0	0	1	–	0

AUF	DB	Aufschlagen eines Datenbausteins	1**/2	0,7	1,2+
	DB-Para.	Aufschlagen eines Datenbausteins (über Parameter adressiert)	2		
	DI	Aufschlagen eines Instanz-Datenbausteins	2		

Statuswort für: **AUF**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

* + Zeit für das Laden der Adresse des Operanden
** bei direkter Adressierung des Operanden

A.38 Baustein-Endeoperationen

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
BE		Beende Baustein	1	2,2
BEA		Beende Baustein absolut	1	2,2

Statuswort für: **BE, BEA**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	0	0	1	–	0

BEB		Beende Baustein bedingt bei VKE=”1”		2,3

Statuswort für: **BEB**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	ja	0	1	1	0

A.39 Tausche Datenbausteine

Tauschen der beiden aktuellen Datenbausteine. Der aktuelle Datenbaustein wird zum aktuellen Instanz-Datenbaustein und umgekehrt. Die Anzeigen werden nicht beeinflusst.

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
TDB		Tausche Datenbausteine	1	0,1

A.40 Sprungoperationen

Sprung, abhängig von der Bedingung. Bei 8-Bit-Operand liegt die Sprungweite zwischen (–128 bis +127). Bei 16-Bit-Operanden liegt die Sprungweite zwischen (–32768 bis –129) oder (+128 bis +32767).

Hinweis zu SPB

Achten Sie bei Programmen für das IM 151-7 CPU darauf, dass das Sprungziel immer der **Beginn** einer Verknüpfungskette ist. Das Sprungziel darf sich nicht innerhalb einer Verknüpfungskette befinden.

Operation	**Operand**	**Bedeutung**	**Länge in Worten**	**typische Ausführungszeit in µs**
SPA	MARKE	Springe unbedingt	1*/2	2,0

Statuswort für: **SPA**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

SPB	MARKE	Springe bedingt bei VKE=”1”	1*/2	2,0
SPBN	MARKE	Springe bedingt bei VKE=”0”	2	2,0

Statuswort für: **SPB, SPBN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	–	0	1	1	0

SPBB	MARKE	Springe bedingt bei VKE=”1” Retten des VKE in das BIE-Bit	2	2,0
SPBNB	MARKE	Springe bedingt bei VKE=”0” Retten des VKE in das BIE-Bit	2	2,0

Statuswort für: **SPBB, SPBNB**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	ja	–	–	–	–	0	1	1	0

SPBI	MARKE	Springe bedingt bei BIE=”1”	2	2,0
SPBIN	MARKE	Springe bedingt bei BIE=”0”	2	2,0

Statuswort für: **SPBI, SPBIN**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	ja	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	0	1	–	0

* 1 Wort lang bei Sprungweiten von –128 bis +127

Operation	Operand	Bedeutung	Länge in Worten	typische Ausführungszeit in µs
SPO	MARKE	Springe bedingt bei Überlauf speichernd (OV="1")	1*/2	2,0

Statuswort für: **SPO**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	ja	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

SPS	MARKE	Springe bedingt bei Überlauf speichernd (OS="1")	2	2,0

Statuswort für: **SPS**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	ja	–	–	–	–
Operation beeinflusst:	–	–	–	–	0	–	–	–	–

SPU	MARKE	Springe bei "Unzulässiger Operation" (A1=1 und A0=1)	2	2,0
SPZ	MARKE	Springe bedingt bei Ergebnis=0 (A1=0 und A0=0)	1*/2	2,0
SPP	MARKE	Springe bedingt bei Ergebnis>0 (A1=1 und A0=0)	1*/2	2,0
SPM	MARKE	Springe bedingt bei Ergebnis<0 (A1=0 und A0=1)	1*/2	2,0
SPN	MARKE	Springe bedingt bei Ergebnis $\neq 0$ (A1=1 und A0=0) oder (A1=0) und (A0=1)	1*/2	2,0
SPMZ	MARKE	Springe bedingt bei Ergebnis ≤ 0 (A1=0 und A0=1) oder (A1=0 und A0=0)	2	2,0
SPPZ	MARKE	Springe bedingt bei Ergebnis ≥ 0 (A1=1 und A0=0) oder (A1=0) und (A0=0)	2	2,0

Statuswort für: **SPU, SPZ, SPP, SPM, SPN, SPMZ, SPPZ**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	ja	ja	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

SPL	MARKE	Sprungverteiler Der Operation folgt eine Liste von Sprungoperationen. Der Operand ist eine Sprungmarke auf die der Liste folgenden Operation. AKKU1-L enthält die Nr. der Sprungoperation, der ausgeführt werden soll	2	2,8
LOOP	MARKE	Dekrementiere AKKU1-L und springe bei AKKU1-L $\neq 0$ (Schleifenprogrammierung)	2	2,2

Statuswort für: **SPL, LOOP**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

* 1 Wort lang bei Sprungweiten von –128 bis +127

A.41 Operationen für das Master Control Relay (MCR)

MCR schreibt den Wert “0” oder lässt Speicherinhalt unverändert.

MCR=0→MCR ist deaktiviert
MCR=1→MCR ist aktiviert; “T”-Operation schreibt Nullen auf die entsprechenden Operanden; “S”-/“R”-Operationen lassen Speicherinhalt unverändert.

Operation	**Operand**	**Bedeutung**	**Länge in Worten**	**typische Ausführungszeit in µs**
MCR(		Öffnen einer MCR-Zone. Retten des VKE auf den MCR-Stack.	1	0,8

Statuswort für: **MCR(**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	ja	–
Operation beeinflusst:	–	–	–	–	–	0	1	–	0

)MCR		Schließen einer MCR-Zone. Entfernen eines Eintrags vom MCR-Stack.	1	0,8

Statuswort für: **)MCR**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	0	1	–	0

MCRA		Aktiviere MCR	1	0,1
MCRD		Deaktiviere MCR	1	0,1

Statuswort für: **MCRA, MCRD**	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation hängt ab von:	–	–	–	–	–	–	–	–	–
Operation beeinflusst:	–	–	–	–	–	–	–	–	–

A.42 Systemzustandsliste (SZL)

Mögliche SZL-Teillisten

In der folgenden Tabelle sind alle möglichen Teillisten mit den zugehörigen Teillistenauszügen und den SZL-IDs dargestellt.

Einzelheiten zum Auslesen der SZL mit beispielsweise der SFC 51 und detaillierte Angaben zum Inhalt finden Sie:

- im Referenzhandbuch *Systemsoftware für S7-300/400* Kapitel *System- und Standardfunktionen Systemzustandsliste SZL* oder
- in der Online-Hilfe zu STEP 7, Kontexthilfe zu den SFBs/SFCs.

Tabelle A-1 SZL-Teillisten des IM151/CPU

Teilliste	SZL-ID Teilliste	Teillistenauszug	SZL-ID Auszug
Baugruppen-Identifikation	W#16#xy11	CPU-Identifikation	W#16#0111
CPU-Merkmale	W#16#xy12	Alle Merkmale Merkmale einer Gruppe Nur SZL-Teillisten-Kopfinfo	W#16#0012 W#16#0112 W#16#0F12
Anwenderspeicherbereiche	W#16#xy13	Datensätze aller Anwenderspeicherbereiche	W#16#0013
Systembereiche	W#16#xy14	Datensätze aller Systembereiche	W#16#0014
Bausteintypen	W#16#xy15	Datensätze aller Bausteintypen Datensätze aller OBs	W#16#0015 W#16#0115
Zustand der Baugruppen-LEDs	W#16#xy19	Zustand aller LEDs Nur SZL-Teillisten-Kopfinfo	W#16#0019 W#16#0F19
Alarmstatus	W#16#xy22	Datensatz zum angegebenen Alarm	W#16#0222
Kommunikation: Zustandsdaten	W#16#xy32	Zustandsdaten zu einem Kommunikationsteil Zustandsdaten zu einem Kommunikationsteil	W#16#0132 W#16#0232
Zustand der Baugruppen-LEDs	W#16#xy74	Zustand aller LEDs Zustand einer LED Nur SZL-Teillisten-Kopfinfo	W#16#0074 W#16#0174 W#16#0F74

Tabelle A-1 SZL-Teillisten des IM151/CPU

Teilliste	SZL-ID Teilliste	Teillistenauszug	SZL-ID Auszug
Baugruppenzustandsinformation	W#16#xy91	Baugruppenzustandsinformation aller gesteckten Module	W#16#0A91
		Baugruppenzustandsinformation einer Baugruppe im zentralen Aufbau oder an einer integrierten DP-Anschaltung	W#16#0C91
		Baugruppenzustandsinformation aller Baugruppen im angegebenen Baugruppenträger/ in der angegebenen DP-Station	W#16#0D91
Baugruppenträger- /Stationszustandsinformation	W#16#xy92	Sollzustand im zentralen Aufbau des Mastersystems	W#16#0092
		Istzustand im zentralen Aufbau des Mastersystems	W#16#0292
		OK-Zustand der Erweiterungsgeräte des Mastersystems	W#16#0692
		Nur SZL-Teillisten-Kopfinfo	W#16#0F92
Diagnosepuffer	W#16#xyA0	Alle im Betriebszustand lieferbaren Einträge	W#16#00A0
		Die neuesten Eintragungen	W#16#01A0
Baugruppendiagnoseinfo (DS 0)	W#16#00B1	–	–
Baugruppendiagnoseinfo (DS 1), physikalische Adresse	W#16#00B2	–	–
Baugruppendiagnoseinfo (DS 1), logische Adresse	W#16#00B3	–	–

Ausführungszeiten der SFCs und SFBs

B

Das IM 151-7 CPU stellt Ihnen verschiedene Systemfunktionen bzw. Systemfunktionsbausteine z. B. zur Programmbearbeitung und Diagnose zur Verfügung. Diese Systemfunktionen/Systemfunktionsbausteine rufen Sie in Ihrem Anwenderprogramm über die Nummer des SFCs auf.

Eine ausführliche Beschreibung aller Systemfunktionen/Systemfunktionsbausteine finden Sie im Referenzhandbuch *STEP 7 System- und Standardfunktionen*. Dort ist beschrieben, wie Sie die Systemfunktionen aufrufen und welche Parameter Sie übergeben müssen.

B.1 Systemfunktionen (SFCs)

SFC-Nr.	Name	Beschreibung	Ausführungszeit in µs
0	SET_CLK	Setzen der Uhrzeit	195
1	READ_CLK	Lesen der Uhrzeit	60
2	SET_RTM	Setzen des Betriebsstundenzählers	65
3	CTRL_RTM	Starten bzw. Stoppen des Betriebsstundenzählers	60
4	READ_RTM	Lesen des Betriebsstundenzählers	90
5	GADR_LGC	Ermitteln der freien Adresse von Kanal *x* der Signalbaugruppe auf Baugruppensteckplatz *y*	135
6	RD_SINFO	Auslesen der Startinformation des aktuellen OB	110
7	DP_PRAL	Auslösen eines Prozessalarms aus dem Anwenderprogramm der CPU als DP-Slave hin zum DP-Master	90
12	D_ACT_DP	Deaktivieren und Aktivieren von DP-Slaves	410
14	DPRD_DAT	Lesen von konsistenten Nutzdaten	150
15	DPWR_DAT	Schreiben von konsistenten Nutzdaten	150
17	ALARM_SQ	Erzeugung quittierbarer bausteinbezogener Meldungen	250
18	ALARM_S	Erzeugung nicht quittierbarer bausteinbezogener Meldungen	250
19	ALARM_SC	Ermittlung des Quittierzustands der letzten ALARM_SQ-Gekommenen-Meldung	110

SFC-Nr.	Name	Beschreibung	Ausführungszeit in µs
20	BLKMOV	Kopieren von Variablen beliebigen Typs **Hinweis:** Die SFC 20 können Sie nur bei ablaufrelevanten DBs anwenden, d. h. bei den Objekteigenschaften ist das Optionsfeld “Unlinked” nicht selektiert. Anderenfalls erhalten Sie als RET_VAL den Fehlercode W#16#8092 (= Baustein nicht ablaufrelevant).	75 +1,6 pro Byte
21	FILL	Vorbesetzen eines Feldes	75 +2,2 pro Byte
22	CREATE_DB	Erzeugen eines Datenbausteins	110 +3,5 pro DB
23	DEL_DB	Löschen eines Datenbausteins	402
24	TEST_DB	Testen eines Datenbausteins	110
28	SET_TINT	Uhrzeitalarm stellen	160
29	CAN_TINT	Uhrzeitalarm stornieren	70
30	ACT_TINT	Uhrzeitalarm aktivieren	120
31	QRY_TINT	Uhrzeitalarm abfragen	75
32	SRT_DINT	Verzögerungsalarm starten	75
33	CAN_DINT	Verzögerungsalarm stornieren	50
34	QRY_DINT	Verzögerungsalarm abfragen	71
36	MSK_FLT	Maskieren von Synchronfehlerereignissen	110
37	DMSK_FLT	Freigeben von Synchronfehlerereignissen	120
38	READ_ERR	Abfragen und Löschen von aufgetretenen und gesperrten Programmier- und Zugriffsfehlerereignissen	120
39	DIS_IRT	Sperren der Bearbeitung neuer Unterbrechungsereignisse	155
40	EN_IRT	Freigeben der Bearbeitung neuer Unterbrechungsereignisse	105
41	DIS_AIRT	Verzögern der Bearbeitung von Unterbrechungsereignissen	45
42	EN_AIRT	Freigeben der Bearbeitung von Unterbrechungsereignissen	45
43	RE_TRIGR	Nachtriggern der Zykluszeitüberwachung	40
44	REPL_VAL	Kopiere Ersatzwert in den AKKU 1 der fehlerverursachenden Ebene	50
46	STP	Überführen der CPU in den STOP-Zustand	keine Angabe
47	WAIT	Realisierung von Wartezeiten	250
49	LGC_GADR	Umrechnung einer freien Adresse in den zugehörigen Steckplatz sowie Baugruppenträger einer Baugruppe	210
50	RD_LGADR	Ermittlung aller vereinbarten freien Adressen einer Baugruppe	420
51	RDSYSST	Auslesen der Informationen aus der Systemzustandsliste. Der SFC 51 ist nicht unterbrechbar durch Alarme.	224 +10 pro Byte
52	WR_USMSG	Eintragen von wählbaren Diagnoseinformationen in den Diagnosepuffer	235

SFC-Nr.	Name	Beschreibung	Ausführungszeit in µs
55	WR_PARM	Schreiben von dynamischen Parametern zu einer Baugruppe (z. Zt. keine Anwendung, da alle Baugruppenparameter statisch sind)	1700
56	WR_DPARM	Schreiben von vordefinierten dynamischen Parametern zu einer Baugruppe (z. Zt. keine Anwendung, da alle Baugruppenparameter statisch sind)	1750
57	PARM_MOD	Parametrieren einer Baugruppe	< 1400
58	WR_REC	Schreiben eines baugruppenspezifischen Datensatzes (z. Zt. keine Anwendung, da kein Modul existiert, zu dem Nutzdatensätze geschrieben werden können)	1400 +32 pro Byte
59	RD_REC	Lesen eines baugruppenspezifischen Datensatzes (z. Zt. nur Lesen der Diagnosedatensätze 0 und 1)	500
64	TIME_TCK	Auslesen der Systemzeit mit einer Genauigkeit von 10 ms	50
65	X_SEND	sicheres Senden von Daten an Teilnehmer außerhalb der S7-Station	310
66	X_RCV	sicheres Empfangen von Daten von Teilnehmern außerhalb der S7-Station	120
67	X_GET	Daten aus einem Teilnehmer außerhalb der S7-Station lesen	190
68	X_PUT	Daten in einen Teilnehmer außerhalb der S7-Station schreiben	190
69	X_ABORT	Abbruch einer bestehenden Verbindung zu einem Teilnehmer außerhalb der S7-Station	100
72	I_GET	Daten aus einem Teilnehmer innerhalb der S7-Station lesen	190
73	I_PUT	Daten in einen Teilnehmer innerhalb der S7-Station schreiben	190
74	I_ABORT	Abbruch einer bestehenden Verbindung zu einem Teilnehmer innerhalb der S7-Station	100
81	UBLKMOV	Variable ununterbrechbar kopieren	75 +2 pro Byte
82	CREA_DBL	Datenbaustein im Ladespeicher erzeugen	< 1050
83	READ_DBL	Lesen aus einem Datenbaustein im Ladespeicher	< 950
84	WRIT_DBL	Schreiben in einen Datenbaustein im Ladespeicher	< 900
101	RTM	Betriebsstundenzähler setzen/starten/stoppen/lesen	
102	RD_DPARA	vordefinierte Parameter lesen	< 1500

B.2 Systemfunktionsbausteine (SFBs)

SFB-Nr.	Name	Beschreibung	Ausführungszeit in µs
IEC-Counter			
0	CTU	Vorwärtszählen. Der Zähler wird bei jeder steigenden Flanke um 1 erhöht.	90
1	CTD	Rückwärtszählen. Der Zähler wird bei jeder steigenden Flanke um 1 erniedrigt.	90
2	CTUD	Vorwärts- und Rückwärtszählen.	100
IEC-Timer			
3	TP	Erzeugen eines Impulses der Zeitdauer PT.	115
4	TON	Verzögern einer steigenden Flanke um die Zeitdauer PT.	101
5	TOF	Verzögern einer fallenden Flanke um die Zeitdauer PT.	100
Realisierung eines Schrittschaltwerks			
32	DRUM	Realisieren eines Schrittschaltwerks mit maximal 16 Schritten.	80
SFBs nach Norm IEC 61784-1:2002 Ed1 CP 3/1			
52	RDREC	Datensatz lesen (dezentral/zentral)	
53	WRREC	Datensatz schreiben (dezentral/zentral)	
54	RALRM	Alarm empfangen (dezentral/zentral)	
75	SALRM	Alarm zum Master senden	

Stellung des IM 151-7 CPU in der CPU-Landschaft

C

In diesem Kapitel zeigen wir Ihnen die wichtigsten Unterschiede zu zwei ausgewählten CPUs der SIMATIC-Familie S7-300.

Außerdem zeigen wir Ihnen, wie Sie Programme, die Sie für S7-300-CPUs geschrieben haben, für das IM 151-7 CPU umschreiben müssen.

Kapitelübersicht

Kapitel	Thema	Seite
C.1	Unterschiede zu ausgewählten S7-300 CPUs	C-2
C.2	Portierung des Anwenderprogrammes	C-3

Weitere Informationen

Weitere Informationen zur Vorgehensweise bei der Erstellung und Strukturierung von Programmen finden Sie in den Handbüchern und der Online-Hilfe zu *STEP 7*.

C.1 Unterschiede zu ausgewählten S7-300 CPUs

In der folgenden Tabelle zeigen wir Ihnen die für die Programmierung wichtigsten Unterschiede zwischen zwei ausgewählten CPUs der SIMATIC-Familie S7-300 und dem IM 151-7 CPU.

Tabelle C-1 Unterschiede zu ausgewählten S7-300 CPUs

Merkmal	CPU 315-2 DP	CPU 315-2 DP (modular)	IM 151-7 CPU	
			(6ES7 151-7Ax00-0AB0)	(6ES7 151-7Ax10-0AB0)
Echtzeituhr	Hardware	Hardware	Software	Hardware
Pufferung des Speichers	ja, Batterie	wird durch MMC gewährleistet (wartungsfrei)	nicht möglich	wird durch MMC gewährleistet (wartungsfrei)
Speicherkarte	Memory-Card	MMC	MMC	MMC
Anzahl der Verbindungen zu PG und OP	4 (ab 10/99: 12)	16	5	max. 12
Einstellung der PROFIBUS-Adresse	Hardware-Konfiguration	Hardware-Konfiguration	Hardware-Konfiguration muss mit Adresseinsteller übereinstimmen	Hardware-Konfiguration
Übertragungsgeschwindigkeit zu PG und OP	187,5 kBaud (MPI) 12 MBaud (DP)	187,5 kBaud (MPI) 12 MBaud (DP)	12 MBaud (DP)	12 MBaud (MPI/DP)
Kommunikation:				
PG/OP	ja	ja	ja	ja
Globaldatenkomm.	ja	ja	nein	ja
S7-Basiskomm.	ja	ja	ja (Server)	ja
S7-Komm.	ja (Server)	ja	ja (Server)	ja (Server)
Direkter Datenaustausch	ja	ja	ja	ja
Einsatzspektrum mit DP	als DP-Master als DP-Slave stand alone	als DP-Master als DP-Slave stand alone	als DP-Slave stand alone	als DP-Slave stand alone
Adressierung	frei	frei	frei	frei
Alarmreaktionszeit	0,4-1,3 ms	0,3-1,2 ms	kleiner 20 ms	kleiner 20 ms
Ziehen/Stecken von Baugruppen im laufenden Betrieb	nein	nein	ja	ja

C.2 Portierung des Anwenderprogrammes

Einführung

Unter Portierung verstehen wir die dezentrale Nutzbarmachung eines Programmes, welches vorher zentral auf einem Master verwendet wurde. Um ein bestehendes Programm ganz oder teilweise vom Master auf einen intelligenten Slave zu verlagern, können bestimmte Anpassungen notwendig werden. Der Aufwand für die Portierung von Teilen eines Anwenderprogrammes auf einen intelligenten Slave hängt davon ab, wie die Adresszuordnung von Ein- und Ausgängen in den FBs im Quellprogramm hinterlegt ist.

Die Ein-und Ausgänge können in den FCs im Quellprogramm auf verschiedene Arten verwendet werden. In der heutigen ET 200S können Adressen gepackt werden, was aber bei dem IM 151-7 CPU nicht möglich ist.

Siehe Adressierung IM 151-7 CPU in Kapitel 3.1.

Portierung bei ungepackten Adressen

Werden FBs mit ungepackten Adressen der E/As verwendet, können die benötigten Teilprogramme problemlos, also ohne Portierungsaufwand in das IM 151-7 CPU übertragen werden.

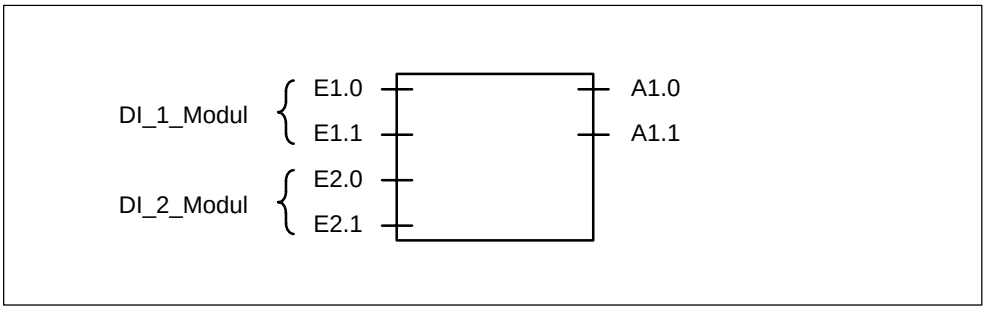


Bild C-1 Beispiel: FB mit ungepackten Adressen

Portierung bei gepackten Adressen

Werden FBs mit gepackten Adressen der E/As in das IM 151-7 CPU kopiert, können dort die gepackten Adressen nicht mehr den E/As der Peripheriemodule vor Ort zugewiesen werden, da die CPU des IM 151-7 CPU nicht mit gepacken Adressen arbeiten kann. In dem Fall ist eine Umverdrahtung des entsprechenden FBs notwendig. Das Umverdrahten entspricht einem “Entpacken” der Adressen.

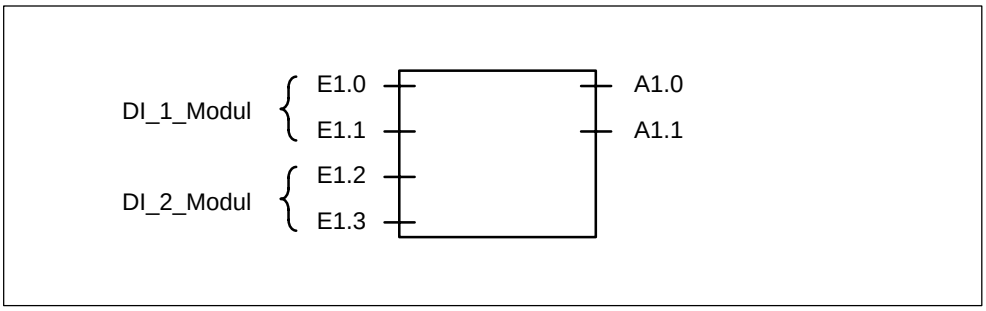


Bild C-2 Beispiel: FB mit gepackten Adressen

Umverdrahten

Grundsätzlich können folgende Bausteine und Operanden umverdrahtet werden:

- Eingänge, Ausgänge
- Merker, Zeiten, Zähler
- Funktionen, Funktionsbausteine

Um die Signale umzuverdrahten, gehen Sie wie folgt vor:

1. Markieren Sie im SIMATIC-Manager den Ordner “Bausteine”, in dem sich die Bausteine mit den gepackten Adressen befinden, die Sie in das IM 151-7 CPU portieren möchten.
2. Wählen Sie den Menübefehl **Extras → Umverdrahten**.
3. Tragen Sie im angezeigten Dialogfeld “Umverdrahten” die gewünschten Ersetzungen (Alter Operand/Neuer Operand) in die Tabelle ein.

Tabelle C-2 Beispiel: Ersetzungen unter Extra → Umverdrahten

	Alter Operand	Neuer Operand
1	E 1.2	E 2.0
2	E 1.3	E 2.1

4. Betätigen Sie die Schaltfläche OK.

Damit starten Sie das Umverdrahten. Nach der Umverdrahtung können Sie über ein Dialogfeld entscheiden, ob Sie die Infodatei zur Umverdrahtung ansehen wollen. Die Infodatei enthält die Operandenliste, “Alter Operand” und “Neuer Operand”. Dazu werden die einzelnen Bausteine aufgeführt mit der Anzahl Verdrahtungen, welche jeweils im Baustein durchgeführt worden sind.

Bild C-3 Beispiel: Umverdrahtung der Signale

Wenn Sie in *STEP 7* den Ein- und Ausgängen Symbole über die Symboltabelle zuweisen, ist in dem Fall ein Ändern der Symboltabelle erforderlich, um auch hier das Teilprogramm für die Verwendung im IM 151-7 CPU anzupassen.

Siehe auch Online-Hilfe von *STEP 7*.

Portierung von FBs mit E/As in einem Peripheriewort

Wenn Sie die Adressen der Ein- und Ausgänge über ein Peripheriewort auf einen von Ihnen programmierten Funktionsbaustein abbilden, ist eine Portierung mit wesentlich mehr Aufwand verbunden.

Eine Möglichkeit besteht in der Programmierung einer Schale um den FB, die eine Anpassung für die Nutzung des FBs mit dem IM 151-7 CPU vornimmt. Die andere Möglichkeit besteht in der Neuprogrammierung des FBs. Wir empfehlen Ihnen die Neuprogrammierung des FBs, da diese Methode leichter zu realisieren ist, als die Programmierung einer Schale.

Siehe auch Online-Hilfe von *STEP 7*.

Glossar

Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden oder Operandenbereich
Beispiele: Eingang E 12.1; Merkerwort MW 25; Datenbaustein DB 3.

AKKU

Die Akkumulatoren sind Register in der → CPU und dienen als Zwischenspeicher für Lade-, Transfer- sowie Vergleichs-, Rechen- und Umwandlungsoperationen.

Alarm

Das Betriebssystem der CPU kennt 10 verschiedene Prioritätsklassen, die die Bearbeitung des Anwenderprogramms regeln. Zu diesen Prioritätsklassen gehören u.a. Alarme, z. B. Diagnosealarme. Bei Auftreten eines Alarms wird vom Betriebssystem automatisch ein zugeordneter Organisationsbaustein aufgerufen, in dem der Anwender die gewünschte Reaktion programmieren kann (z. B. in einem FB).

Alarm, Diagnose- → Diagnosealarm

Alarm, Prozess- → Prozessalarm

ANLAUF

Der Betriebszustand ANLAUF wird beim Übergang vom Betriebszustand STOP in den Betriebszustand RUN durchlaufen.

Kann ausgelöst werden durch den Betriebsartenschalter oder nach Netz-Ein oder durch Bedienung am Programmiergerät. Bei ET 200S wird ein Neustart durchgeführt.

Anwenderprogramm

Bei SIMATIC wird unterschieden zwischen → Betriebssystem der CPU und Anwenderprogrammen. Letztere werden mit der Programmiersoftware → *STEP 7* in den möglichen Programmiersprachen (Kontaktplan und Anweisungsliste) erstellt und sind in Codebausteinen gespeichert. Daten sind in Datenbausteinen gespeichert.

Anwenderspeicher

Der Anwenderspeicher enthält Code- und Datenbausteine des Anwenderprogramms. Der Anwenderspeicher kann sowohl in der CPU integriert sein oder auf zusteckbaren Memory Cards (IM 151-7 CPU) bzw. Speichermodulen. Das Anwenderprogramm wird jedoch grundsätzlich aus dem → Arbeitsspeicher der CPU abgearbeitet.

Arbeitsspeicher

Der Arbeitsspeicher ist ein RAM-Speicher in der → CPU, auf den der Prozessor während der Bearbeitung des Anwenderprogramms zugreift.

Automatisierungssystem

Ein Automatisierungssystem ist eine speicherprogrammierbare Steuerung, die aus mindestens einer CPU, verschiedenen Ein- und Ausgabebaugruppen sowie Bedien- und Beobachtungsgeräten besteht.

Backup-Speicher

Der Backup-Speicher gewährleistet eine Pufferung von Speicherbereichen der → CPU ohne Pufferbatterie. Gepuffert wird eine parametrierbare Anzahl von Zeiten, Zählern, Merkern und Datenbytes, die remanenten Zeiten, Zähler, Merker und Datenbytes.

Baudrate

Die Baudrate ist die Geschwindigkeit bei der Datenübertragung und gibt die Anzahl der übertragenen Bits pro Sekunde an (Baudrate = Bitrate).

Bei ET 200S sind Baudraten von 9,6 kBaud bis 12 MBaud möglich.

Betriebssystem der CPU

Das Betriebssystem der CPU organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer speziellen Steuerungsaufgabe verbunden sind.

Betriebszustand

Die Automatisierungssysteme von SIMATIC S7 kennen folgende Betriebszustände: STOP, → ANLAUF, RUN.

Bus

gemeinsamer Übertragungsweg, mit dem alle Teilnehmer verbunden sind; besitzt zwei definierte Enden

Bei ET 200S ist der Bus eine Zweidrahtleitung oder ein Lichtwellenleiter.

CPU

Central Processing Unit = Zentralbaugruppe des S7-Automatisierungssystems mit Steuer- und Rechenwerk, Speicher, Betriebssystem und Schnittstelle für Programmiergerät.

Datenbaustein

Datenbausteine (DB) sind Datenbereiche im Anwenderprogramm, die Anwenderdaten enthalten. Es gibt globale Datenbausteine, auf die von allen Codebausteinen zugegriffen werden kann und es gibt Instanzdatenbausteine, die einem bestimmten FB-Aufruf zugeordnet sind.

Datenquerverkehr

siehe Direkter Datenaustausch

Dezentrale Peripheriesysteme

sind Ein-/Ausgabeeinheiten, die nicht im Zentralgerät eingesetzt werden, sondern dezentral in größerer Entfernung von der CPU aufgebaut sind, z. B.:

- ET 200S, ET 200M, ET 200B, ET 200C, ET 200U, ET 200X, ET 200L
- DP/AS-i Link
- S5-95U mit PROFIBUS-DP-Slave-Schnittstelle
- weitere DP-Slaves der Fa. Siemens oder weiterer Hersteller

Die dezentralen Peripheriesysteme sind über PROFIBUS-DP mit dem DP-Master verbunden.

Diagnose

Diagnose ist die Erkennung, Lokalisierung, Klassifizierung, Anzeige, weitere Auswertung von Fehlern, Störungen und Meldungen.

Diagnose bietet Überwachungsfunktionen, die während des Anlagenbetriebs automatisch ablaufen. Dadurch erhöht sich die Verfügbarkeit von Anlagen durch Verringerung der Inbetriebsetzungszeiten und Stillstandszeiten.

Diagnosealarm

Diagnosefähige Baugruppen melden erkannte Systemfehler über Diagnosealarme an die zentrale CPU.

In SIMATIC S7/M7: Bei Erkennen bzw. bei Verschwinden eines Fehlers (z. B. Drahtbruch) löst ET 200S bei freigegebenem Alarm einen Diagnosealarm aus. Die CPU des DP-Masters unterbricht die Bearbeitung des Anwenderprogramms bzw. niederpriorer Prioritätsklassen und bearbeitet den Diagnosealarmbaustein (OB 82).

In SIMATIC S5: Der Diagnosealarm wird innerhalb der gerätebezogenen Diagnose nachgebildet. Durch zyklisches Abfragen der Diagnosebits in der gerätebezogenen Diagnose können Sie Fehler wie z. B. Drahtbruch erkennen.

Diagnosepuffer

Der Diagnosepuffer ist ein gepufferter Speicherbereich in der CPU, in dem Diagnoseereignisse in der Reihenfolge des Auftretens abgelegt sind.

Direkter Datenaustausch

Direkter Datenaustausch ist eine spezielle Kommunikationsbeziehung zwischen PROFIBUS-DP-Teilnehmern. Der direkte Datenaustausch ist dadurch gekennzeichnet, dass PROFIBUS-DP-Teilnehmer “mithören”, welche Daten ein DP-Slave seinem DP-Master zurückschickt.

DP-Master

Ein → Master, der sich nach der Norm IEC 61784-1:2002 Ed1 CP 3/1 verhält, wird als DP-Master bezeichnet.

DP-Norm

DP-Norm ist das Busprotokoll des Dezentralen Peripheriesystems ET 200 nach der Norm IEC 61784-1:2002 Ed1 CP 3/1.

DP-Slave

Ein → Slave, der am PROFIBUS mit dem Protokoll PROFIBUS-DP betrieben wird und sich nach der Norm IEC 61784-1:2002 Ed1 CP 3/1 verhält, heißt DP-Slave.

ET 200

Das Dezentrale Peripheriesystem ET 200 mit dem Protokoll PROFIBUS-DP ist ein Bus zum Anschluss von dezentraler Peripherie an eine CPU oder einen adäquaten DP-Master. ET 200 zeichnet sich durch schnelle Reaktionszeiten aus, da nur wenige Daten (Bytes) übertragen werden.

ET 200 basiert auf der Norm IEC 61784-1:2002 Ed1 CP 3/1.

ET 200 arbeitet nach dem Master-Slave-Prinzip. DP-Master können z. B. die Masteranschaltung IM 308-C oder die CPU 315-2 DP sein.

DP-Slaves können die dezentrale Peripherie ET 200S, ET 200B, ET 200C, ET 200M, ET 200X, ET 200U, ET 200L oder DP-Slaves der Fa. Siemens oder weiterer Hersteller sein.

FC → Funktion

Fehleranzeige

Die Fehleranzeige ist eine der möglichen Reaktionen des Betriebssystems auf einen Laufzeitfehler. Die anderen Reaktionsmöglichkeiten sind: → Fehlerreaktion im Anwenderprogramm, STOP-Zustand der CPU.

Fehlerbehandlung über OB

Erkennt das Betriebssystem einen bestimmten Fehler (z.B. Zugriffsfehler bei *STEP 7*), so ruft es den für diesen Fall vorgesehenen Organisationsbaustein (Fehler-OB) auf, in dem das weitere Verhalten der CPU festgelegt werden kann.

Fehlerreaktion

Reaktion auf einen → Laufzeitfehler. Das Betriebssystem kann auf folgende Arten reagieren: Überführen des Automatisierungssystems in den STOP-Zustand, Aufruf eines Organisationsbausteins, in dem der Anwender eine Reaktion programmieren kann oder Anzeigen des Fehlers.

FORCEN

Mit der Funktion “Forcen” lassen sich z.B. während der Inbetriebnahmephase bestimmte Ausgänge auch bei fehlender Erfüllung logischer Verknüpfungen des Anwenderprogramms (z.B. durch fehlende Verdrahtung von Eingängen) für beliebig lange Zeiträume auf Zustand “EIN” setzen.

FREEZE

ist ein Steuerkommando des DP-Masters an eine Gruppe von DP-Slaves.

Nach Erhalt des Steuerkommandos FREEZE friert der DP-Slave den aktuellen Zustand der **Eingänge** ein und überträgt diese zyklisch an den DP-Master.

Nach jedem neuen Steuerkommando FREEZE friert der DP-Slave erneut den Zustand der **Eingänge** ein.

Die Eingangsdaten werden erst dann wieder zyklisch vom DP-Slave an den DP-Master übertragen, wenn der DP-Master das Steuerkommando UNFREEZE sendet.

Funktion

Eine Funktion (FC) ist gemäß IEC 1131-3 ein Codebaustein ohne statische Daten. Eine Funktion bietet die Möglichkeit der Übergabe von Parametern im Anwenderprogramm. Dadurch eignen sich Funktionen zur Programmierung von häufig wiederkehrenden komplexen Funktionen, z.B. Berechnungen.

GSD-Datei

In einer GSD-Datei (Geräte-Stammdaten-Datei) sind alle DP-slavespezifischen Eigenschaften hinterlegt. Das Format der GSD-Datei ist in der Norm IEC 61784-1:2002 Ed1 CP 3/1 hinterlegt.

Intelligenter DP-Slave

Merkmal eines intelligenten DP-Slaves ist, dass Ein-/Ausgangsdaten nicht unmittelbar von einem realen Ein-/Ausgang des DP-Slaves dem DP-Master zur Verfügung gestellt werden, sondern von einer vorverarbeitenden CPU, hier vom Interfacemodul IM 151-7 CPU.

Komprimieren

Mit der PG-Online-Funktion “Komprimieren” werden alle gültigen Bausteine im RAM der CPU bündig und lückenlos an den Anfang des Anwenderspeichers geschoben. Dadurch verschwinden alle Lücken, die beim Löschen oder Korrigieren von Bausteinen entstanden sind.

Konsistente Daten

Daten, die inhaltlich zusammengehören und nicht getrennt werden dürfen, bezeichnet man als konsistente Daten.

Zum Beispiel müssen die Werte von Analogbaugruppen immer konsistent behandelt werden, d. h., der Wert einer Analogbaugruppe darf durch das Auslesen zu zwei verschiedenen Zeitpunkten nicht verfälscht werden.

Ladespeicher

Der Ladespeicher ist Bestandteil der CPU. Er beinhaltet vom Programmiergerät erzeugte Objekte. Er ist entweder als zusteckbare Memory Card/Micro Memory Card oder als fest integrierter Speicher realisiert.

Laufzeitfehler

Fehler, die während der Bearbeitung des Anwenderprogramms im Automatisierungssystem (also nicht im Prozess) auftreten.

Masse

Als Masse gilt die Gesamtheit aller untereinander verbundenen inaktiven Teile eines Betriebsmittels, die auch im Fehlerfall keine gefährliche Berührungsspannung annehmen können.

Master

dürfen, wenn sie im Besitz des Tokens sind, Daten an andere Teilnehmer schikken und von anderen Teilnehmern Daten anfordern (= aktiver Teilnehmer). DP-Master ist z.B. die CPU 315-2 DP.

Mastersystem

Alle DP-Slaves, die einem DP-Master lesend und schreibend zugeordnet sind, bilden zusammen mit dem DP-Master ein Mastersystem.

Merker

Merker sind Bestandteil des → Systemspeichers der CPU zum Speichern von Zwischenergebnissen. Auf sie kann bit-, byte-, wort- oder doppelwortweise zugegriffen werden.

MMC

Micro Memory Card. Speichermodul für SIMATIC-Systeme. Einsetzbar als transportabler Datenträger und Ladespeicher.

MPI

Die Mehrpunktfähige Schnittstelle (MPI) ist die Programmiergeräte-Schnittstelle von SIMATIC S7.

Neustart

Beim Anlauf einer CPU (z. B. nach Betätigung des Betriebsartenschalters von STOP auf RUN oder bei Netzspannung EIN) wird vor der zyklischen Programmbearbeitung (OB 1) zunächst der Organisationsbaustein OB 100 (Neustart) bearbeitet. Bei Neustart wird das Prozessabbild der Eingänge eingelesen und das *STEP 7*- Anwenderprogramm beginnend beim ersten Befehl im OB 1 bearbeitet.

OB → Organisationsbaustein

OB-Priorität

Das Betriebssystem der CPU unterscheidet zwischen verschiedenen Prioritätsklassen, z.B. zyklische Programmbearbeitung, prozessalarmgesteuerte Programmbearbeitung. Jeder Prioritätsklasse sind → Organisationsbausteine (OB) zugeordnet, in denen der S7-Anwender eine Reaktion programmieren kann. Die OBs haben standardmäßig verschiedene Prioritäten, in deren Reihenfolge sie im Falle eines gleichzeitigen Auftretens bearbeitet werden bzw. sich gegenseitig unterbrechen.

Organisationsbaustein

Organisationsbausteine (OBs) bilden die Schnittstelle zwischen dem Betriebssystem der CPU und dem Anwenderprogramm. In den Organisationsbausteinen wird die Reihenfolge der Bearbeitung des Anwenderprogramms festgelegt.

Parameter

1. Variable eines *STEP 7*-Codebausteins

2. Variable zur Einstellung des Verhaltens einer Baugruppe (eine oder mehrere pro Baugruppe). Jede Baugruppe besitzt im Lieferzustand eine sinnvolle Grundeinstellung, die durch Konfigurieren in *STEP 7* verändert werden kann.

PG → Programmiergerät

Prioritätsklasse

Das Betriebssystem einer S7-CPU bietet maximal 26 Prioritätsklassen (bzw. “Programmbearbeitungsebenen”), denen verschiedene Organisationsbausteine zugeordnet sind. Die Prioritätsklassen bestimmen, welche OBs andere OBs unterbrechen. Umfasst eine Prioritätsklasse mehrere OBs, so unterbrechen sie sich nicht gegenseitig, sondern werden sequentiell bearbeitet.

PROFIBUS

Process **Fi**eld **Bus**, Prozess- und Feldbusnorm, die in der Norm IEC 61784-1:2002 Ed1 CP 3/1 festgelegt ist. Sie gibt funktionelle, elektrische und mechanische Eigenschaften für ein bitserielles Feldbussystem vor.

PROFIBUS gibt es mit den Protokollen DP (= Dezentrale Peripherie), FMS (= Fieldbus Message Specification), PA (= Prozess-Automation) oder TF (= Technologische Funktionen).

PROFIBUS-Adresse

Jeder Busteilnehmer muss zur eindeutigen Identifizierung am PROFIBUS eine PROFIBUS-Adresse erhalten.

PC/PG haben die PROFIBUS-Adresse “0”.

Für das Dezentrale Peripheriesystem ET 200S sind die PROFIBUS-Adressen 1 bis 125 zulässig.

Programmiergerät

Programmiergeräte sind im Kern Personal Computer, die industrietauglich, kompakt und transportabel sind. Sie sind gekennzeichnet durch eine spezielle Hardware- und Software-Ausstattung für speicherprogrammierbare Steuerungen SIMATIC.

Prozessabbild

Das Prozessabbild ist Bestandteil des → Systemspeichers der CPU. Am Anfang des zyklischen Programmes werden die Signalzustände der Eingänge zum Prozessabbild der Eingänge übertragen. Am Ende des zyklischen Programmes wird das Prozessabbild der Ausgänge als Signalzustand zu den Ausgängen übertragen.

Prozessalarm

Ein Prozessalarm wird ausgelöst von alarmauslösenden Baugruppen aufgrund eines bestimmten Ereignisses im Prozess. Der Prozessalarm wird der CPU gemeldet. Entsprechend der Priorität dieses Alarms wird dann der zugeordnete → Organisationsbaustein bearbeitet.

Publisher

Ein Sender im direkten Datenaustausch. *Siehe* Direkter Datenaustausch

Querverkehr

siehe Direkter Datenaustausch

Schachtelungstiefe

Mit Bausteinaufrufen kann ein Baustein aus einem anderen heraus aufgerufen werden. Unter Schachtelungstiefe versteht man die Anzahl der gleichzeitig aufgerufenen Codebausteine.

SFC → Systemfunktion

Slave

Ein Slave darf nur nach Aufforderung durch einen → Master Daten mit diesem austauschen. Slaves sind z. B. alle DP-Slaves wie ET 200S, ET 200B, ET 200X, ET 200M, usw.

Speicherprogrammierbare Steuerung

Speicherprogrammierbare Steuerungen (SPS) sind elektronische Steuerungen, deren Funktion als Programm im Steuerungsgerät gespeichert ist. Aufbau und Verdrahtung des Gerätes hängen also nicht von der Funktion der Steuerung ab. Die speicherprogrammierbare Steuerung hat die Struktur eines Rechners; sie besteht aus → CPU (Zentralbaugruppe) mit Speicher, Ein-/Ausgabebaugruppen und internem Bus-System. Die Peripherie und die Programmiersprache sind auf die Belange der Steuerungstechnik ausgerichtet.

SPS → Speicherprogrammierbare Steuerung

Stand-alone-Betrieb

Das Gerät läuft autark ohne Datenverkehr zu einem übergeordneten Master und ohne direkten Datenverkehr zu anderen DP-Slaves. Alle Module laufen mit Defaultparametern und mit der max. Konfiguration (32 Slots, 64 Byte konsistent) hoch.

Startereignis

Startereignisse sind definierte Ereignisse wie Fehler, Zeitpunkte oder Alarme. Sie veranlassen das Betriebssystem, einen zugehörigen Organisationsbaustein zu starten (falls vom Anwender programmiert). Die Startereignisse werden in der Kopfinformation des zugehörigen OBs angezeigt. Der Anwender kann im Anwenderprogramm auf Startereignisse reagieren.

STEP 7

Programmiersprache zur Erstellung von Anwenderprogrammen für SIMATIC S7-Steuerungen.

Subscriber

Ein Empfänger im direkten Datenaustausch. *Siehe* Direkter Datenaustausch

Summenstrom

Summe der Ströme aller Ausgangskanäle einer Digital-Ausgabebaugruppe.

SYNC

ist ein Steuerkommando des DP-Masters an eine Gruppe von DP-Slaves.

Mit dem Steuerkommando SYNC veranlasst der DP-Master den DP-Slave, dass der DP-Slave die Zustände der **Ausgänge** auf den momentanen Wert einfriert. Bei den folgenden Telegrammen speichert der DP-Slave die Ausgangsdaten, die Zustände der Ausgänge bleiben aber unverändert.

Nach jedem neuen Steuerkommando SYNC setzt der DP-Slave die Ausgänge, die er als Ausgangsdaten gespeichert hat. Die Ausgänge werden erst dann wieder zyklisch aktualisiert, wenn der DP-Master das Steuerkommando UNSYNC sendet.

Systemfunktion

Eine Systemfunktion (SFC) ist eine im Betriebssystem der CPU integrierte Funktion, die bei Bedarf im *STEP 7*-Anwenderprogramm aufgerufen werden kann.

Systemdiagnose

Systemdiagnose ist die Erkennung, Auswertung und die Meldung von Fehlern, die innerhalb des Automatisierungssystems auftreten. Beispiele für solche Fehler sind: Programmfehler oder Ausfälle auf Baugruppen. Systemfehler können mit LED-Anzeigen oder in *STEP 7* angezeigt werden.

Systemspeicher

Der Systemspeicher ist auf der Zentralbaugruppe integriert und als RAM-Speicher ausgeführt. Im Systemspeicher sind die Operandenbereiche (z. B. Zeiten, Zähler, Merker) sowie vom Betriebssystem intern benötigte Datenbereiche (z. B. Puffer für Kommunikation) abgelegt.

Teilnehmer

Gerät, welches Daten über den Bus senden, empfangen oder verstärken kann, z. B. DP-Master, DP-Slave, RS 485-Repeater, Aktiver Sternkoppler.

Timer → Zeiten

Token

Zugriffsberechtigung am Bus

Zähler

Zähler sind Bestandteile des → Systemspeichers der CPU. Der Inhalt der “Zählerzellen” kann durch *STEP 7*-Anweisungen verändert werden (z. B. vorwärts/ rückwärts zählen).

Zeiten

Zeiten sind Bestandteile des → Systemspeichers der CPU. Asynchron zum Anwenderprogramm wird der Inhalt der “Zeitzellen” automatisch vom Betriebssystem aktualisiert. Mit *STEP 7*-Anweisungen wird die genaue Funktion der Zeitzelle (z. B. Einschaltverzögerung) festgelegt und ihre Bearbeitung (z. B. Starten) angestoßen.

Zykluskontrollpunkt

Abschnitt der CPU-Programmbearbeitung, an dem z. B. das Prozessabbild aktualisiert wird.

Zykluszeit

Die Zykluszeit ist die Zeit, die die → CPU für die einmalige Bearbeitung des → Anwenderprogramms benötigt.

Index

Zeichen

), A-15
)MCR, A-48
+, A-35
+AR1, A-36
+AR2, A-36
+D, A-34
+I, A-33
+R, A-35
–D, A-34
–I, A-33
–R, A-35
*D, A-34
*I, A-33
*R, A-35
/D, A-34
/I, A-33
/R, A-35
=, A-22
==D, A-37
==I, A-36
==R, A-38
<=D, A-37
<=I, A-36
<=R, A-38
<>D, A-37
<>I, A-36
<> R, A-38
<D, A-37
<I, A-36
<R, A-38
>=D, A-37
>=R, A-38
>>=I, A-36
>>D, A-37
>>I, A-36
>>R, A-38

A

Adreßbereich
  Datenkonsistenz, 3-6
  Defaulteinstellung, 3-9
  der Erweiterungsmodule, 3-3
  für Nutzdatentransfer, 3-5
Adresse, Glossar-1
Adressen
  Basisadresse, 3-4
  für Diagnose, 3-6
  für Nutzdatentransfer, 3-5
Adressierung, 3-1
  Adressvergabe, 3-4
  frei, 3-4
  indirekte, A-7
  Oberfläche in STEP 7, 3-8
  Regeln, 3-8
  steckplatzorientiert, 3-2
Adressierungsbeispiele, STEP 7-Operationen, A-5
Adreßregister, A-4
Adreßzuordnung, für Analog- und Digitalmodule, 3-3
Akku, **A-3**, Glossar-1
Alarm, Glossar-1
  Prozeß-, Glossar-10
Alarme, Zyklusverlängerung, 8-4
Allgemeine Technische Daten, 9-2
Anlauf, Glossar-1
  OB 100, 7-37
Anlaufverhalten, 7-39
Anwenderprogramm, Glossar-2
  Hochladen, 7-16
  Laden, 7-15
Anwenderprogrammbearbeitungszeit, 8-2
Anwenderspeicher, Glossar-2
Anwendungsgebiet, 1-2
Anzeige, LED, 6-9
Arbeitsspeicher, 7-13, Glossar-2
AUF, A-44
Aufbau
  ET 200S stand-alone, 4-5
  MPI-Netz, 5-2
  PROFIBUS-Netz, 4-2
Aufbau des Diagnosetelegramms, 6-14
Ausgänge, Verzögerungszeit, 8-4
Automatisierungssystem, Glossar-2

B

Backup-Speicher, Glossar-2
Baudrate, Glossar-2
Baugruppenklassen, Kennung, 6-27
Bausteine
  der IM 151-7 CPU, 7-36
  Hochladen, 7-16
  Laden, 7-16
  Löschen, 7-16
BE, A-45
BEA, A-45
Bearbeitungszeit
  Betriebssystem, 8-3
  Anwenderprogramm, 8-2, **8-4**
  Prozeßabbild-Aktualisierung, 8-3
BEB, A-45
Bestellnummer
  IM 151-7 CPU, 9-2
  Netzkomponenten, 4-6
Betriebsartenschalter, 7-4
  MRES, 7-4
  RUN, 7-4
  STOP, 7-4
  Urlöschen, 6-4
Betriebssystem
  Bearbeitungszeit, 8-3
  der CPU, Glossar-2
Betriebszustand, Glossar-3
  RUN, 7-5
  STOP, 7-5
Betriebszustandsänderungen, 6-12
BLD, A-41
BTD, A-42
BTI, A-42
Bus, Glossar-3
Busanschlußstecker, 4-6

C

CALL, A-44
CC, A-44
CLR, A-23
CPU, Glossar-3
  Betriebssystem, Glossar-2

D

Daten, konsistente, Glossar-7
Datenaustausch
  Beispielprogramm, 3-9
  direkter, 4-12
  mit DP-Master, 3-5
  Prinzip, 3-1
Datenaustausch, direkter, Glossar-4
Datenbaustein, Glossar-3
Datenkonsistenz, **3-6**, 3-7
Datenquerverkehr, Glossar-3
DBs, 7-36
DEC, A-41
Defaultadressierung, 3-2
Defaulteinstellung, für Adreßbereich, 3-9
Dezentrales Peripheriegerät, Glossar-3
Diagnose, 6-1, **Glossar-4**
  gerätebezogen, **6-22**
  kanalspezifische, 6-29
  kennungsbezogen, 6-19
  System-, Glossar-12
  über LED, 6-9
Diagnoseadresse, 3-6, **6-11**, 6-13
Diagnosealarm, **6-23**, Glossar-4
  OB 82, 7-37
Diagnosealarmreaktionszeit, 8-8
Diagnosedaten, 6-25
Diagnosepuffer, Glossar-4
  auslesen, 6-5
  Eintrag, 6-11
Direkter Datenaustausch, 4-12, Glossar-4
DP-Master, Glossar-4
DP-Diagnoseadresse, 3-6
DP-Slave, Glossar-5
  intelligenter, 6-2, **Glossar-6**
DP-Slave-Diagnose, Aufbau, 6-26
DP-Norm, Glossar-4
DTB, A-42
DTR, A-42

E

Eigenschaften, 9-2
  von IM 151-7 CPU, **1-4**, 7-2
Eingänge, Verzögerungszeit, 8-4
Ereigniserkennung, im DP-Master/DP-Slave, 6-12
ET 200, Glossar-5
ET 200S
  Handbücher, 1-5
  Komponenten, 1-5

ET 200S-Handbücher, Wegweiser, 1-5

F

FBs, 7-36
FCs, 7-36
Fehler, Peripherie-Direktzugriff, OB 122, 7-38
Fehleranzeige, Glossar-5
Fehlerbehandlung über OB, Glossar-5
Fehlerreaktion, **7-38**, Glossar-5
Fehlersuche, 6-1
FN, A-21
Forcen, 7-5, Glossar-6
Formatieren der MMC, 7-8
FP, A-21
FR, **A-24**
FR , A-25
FRCE, LED, 7-5
FREEZE, Glossar-6
Freie Adressierung der Peripheriemodule, 3-4
Funktion, FC, Glossar-6
Funktionen, über PG, 4-9

G

Gerätebezogene Diagnose, **6-22**
Globale Datenkommunikation, 7-34
GSD-Datei, **7-2**, Glossar-6
Gültigkeitsbereich, des Handbuchs, iii

H

Handbuch, Zweck, iii
Handbücher
  Wegweiser, 1-5
  weitere, vi
Handbuchinhalte, Kurzübersicht, 1-5
Hardware-Uhr, 7-39
Herstellerkennung, CPU 31x-2 als DP-Slave, 6-18
Hochladen, 7-16

I

IM 151-7 CPU
  Bausteine, 7-36
  Betriebsartenschalter, 7-4
  Eigenschaften, 1-4
  Parameter, 7-39
  projektieren, 6-2
  urlöschen, 6-4
  wichtige Eigenschaften, 7-2
Inbetriebnahme, 2-1, 2-2, 2-6, 2-11, 2-20, 6-1, 6-7
INC, A-41
Inhalte des Handbuchs, v
Integrierte Uhr, 7-31
Intelligenter DP-Slave, 6-2, **Glossar-6**
Interfacemodul IM 151-7 CPU, 9-4
  Prinzipschaltbild, 9-4
  Technische Daten, 9-5
Interfacemodul IM 151-7 CPU FO, 9-3, 9-4
  Anschlußbelegung, 9-3
  Prinzipschaltbild, 9-4
INVD, A-43
INVI, A-43
ITB, A-42
ITD, A-42

K

Kabel, 4-6
kanalspezifische Fehler, 6-29
Kennungsbezogene Diagnose, 6-19
Kommunikation
  Dienste der CPUs, 7-32
  Globale Datenkommunikation, 7-34
  OP-Kommunikation, 7-33
  PG-Kommunikation, 7-32
  S7-Basiskommunikation, 7-33
  S7-Kommunikation, 7-33
Komponenten, ET 200S, 1-5
Komprimieren, 7-17, Glossar-6
Konfigurationsdaten, Übernahme, 7-40

Konfigurieren, 2-1, 2-2, 2-7, 2-12
Konsistente Daten, Glossar-7
Konsistenz, **3-6**, 3-7

L

L, **A-26**, A-27, A-28, A-32
Laden
  Anwenderprogramm, 7-15
  von Bausteinen, 7-16
Ladeoperation, 3-7
Ladespeicher, 7-12, Glossar-7
LAR1, A-31
LAR2, A-31
Laufzeitfehler, Glossar-7
LC, A-28
Lebensdauer einer MMC, 7-6
LED, 6-5
  Anzeige, 1-4, **6-9**
  FRCE, 7-5
  ON, 7-5
  RUN, 7-5
  SF, 7-5
  STOP, 7-5
Lieferpaket, v
Lokaldaten, 7-22
LOOP, A-47
Löschen von Bausteinen, 7-16

M

Masse, Glossar-7
Master, Glossar-7
Master-PROFIBUS-Adresse, 6-17
Mastersystem, Glossar-7
MCR(, A-48
MCRA, A-48
MCRD, A-48
Merker, Glossar-7
Micro Memory Card, 7-6, Glossar-8
MMC, Glossar-8
MMC formatieren, 7-8
MMC-Lebensdauer, 7-6
MMC-Modul, 7-6
MOD, A-34
Modulstatus, 6-20
MPI, 4-2, 5-2, Glossar-8
PROFIBUS-DP, Daten, 7-2
MPI-Schnittstelle, 7-29
MPI-Teilnehmeradresse, 7-40
MRES, Betriebsartenschalter, 7-4

N

NEGD, A-43
NEGI, A-43
MPI-Netz, Prinzipieller Aufbau, 5-2
Netz, Aufbau, 4-1
PROFIBUS-Netz
  Netzkomponenten, 4-6
  Prinzipieller Aufbau, 4-2
Netzkomponenten, 4-6
Neustart, 7-19, Glossar-8
NOP, A-41
Normen, iv
NOT, A-23
Nutzdatentransfer, zum DP-Master, 3-5

O

O, **A-13**, A-16, A-19
O(, A-15
OB 1, 7-37
OB 100, 7-37
OB 122, 6-12, **7-38**
OB 40, **7-37**
OB 82, 6-8, 6-12, **7-37**
OB 86, 6-8, 6-12, **7-38**
OB 87, **7-38**
OB, Glossar-8
  Startereignis, Glossar-11
OB-Priorität, Glossar-8
OB10, 7-37
OB121, 7-38
OB20, 7-37
OB35, 7-37
OB80, 7-38
OB83, 7-37
OB85, 7-38
OBs
  der CPU, 7-36
  Startereignisse, 7-37
OD, A-18
ON, **A-13**, A-17, A-20
  LED, 7-5
ON(, A-15
Online-Funktionen, für ET 200S, 4-9
OP-Kommunikation, 7-33
Operanden, A-2
Operandenbereiche, 7-20
Organisationsbaustein, Glossar-8
OW, A-18

P

Parameter, Glossar-8
IM 151-7 CPU, 7-39
Parameterbereiche, STEP 7-Operationen, A-2
Parametriertelegramm, Aufbau, 7-41
PC
Anschluß an ET 200X, 4-5
Voraussetzungen, 4-2, 5-2
Peripherie-Direktzugriffsfehler, OB 122, 7-38
PG, Glossar-9
Anschluß an ET 200X, 4-5
Steckleitung, 4-4
Voraussetzungen, 4-2, 5-2
PG-Funktionen, 4-9
PG-Kommunikation, 7-32
PG-Steckleitung, 4-6
Pointer, berechnen, A-6
POP, A-41
Priorität, OB, Glossar-8
Prioritätsklasse, Glossar-9
Probelauf, 2-1, 2-2, 2-20
Probelauf , 2-10
PROFIBUS, Glossar-9
PROFIBUS-Adresse, 4-3, **Glossar-9**
PROFIBUS-DP-Schnittstelle, 7-29
Programmieren, 2-1, 2-2, 2-9, 2-17
Programmiersoftware, 1-4
Projektieren, IM 151-7 CPU, 6-2
Projektiersoftware, 1-4
Prommen, 7-17
Prozeßabbild, Glossar-9
Prozessabbild der Ein- und Ausgänge, 7-21
Anwenderprogramm, Bearbeitungszeit, 8-4
Prozeßabbild-Aktualisierung, Bearbeitungszeit, 8-3
Prozeßalarm, 6-23, Glossar-10
OB 40, 7-37
Prozeßalarmreaktionszeit, 8-8
Prozeßalarmverarbeitung, 8-8
Publisher, Glossar-10
PUSH, A-41

Q

Querverkehr, Glossar-10
Querverkehr, siehe Direkter Datenaustausch, 4-12

R

R, **A-22**, A-24, A-25
RAM to ROM, 7-17
Reaktionszeit, 8-5
Diagnosealarm-, 8-8
kürzeste, 8-6
längste, 8-7
Prozeßalarm-, 8-8
Regeln, für Adressierung, 3-8
Remanenter Speicher, 7-13
Remanenzverhalten der Speicherobjekte, 7-14
Remanenz, 7-39
RLD, A-40
RLDA, A-40
RND, A-42
RND+, A-42
RND–, A-42
RRD, A-40
RRDA, A-40
RUN
Betriebsartenschalter, 7-4
Betriebszustand, 7-5
LED, 7-5

S

S, **A-22**, A-25
T, **A-29**, A-30, A-32
S7-Basiskommunikation, 7-33
S7-Kommunikation, 7-33
SA, A-24
SAVE, A-23
Schachtelungstiefe, Glossar-10
Schnittstellen
MPI-Schnittstelle, 7-29
PROFIBUS-DP-Schnittstelle, 7-29
Welche Geräte an welche Schnittstelle?, 7-30
SE, A-24
SET, A-23
SF, LED, 7-5
SFBs, 7-36
SFC, B-1
SFC DPRD_DAT, 3-7
SFC DPWR_DAT, 3-7
SFCs, 7-36
Ausführungszeiten, B-1
SI, A-24
SIMATIC Micro Memory Card, 7-6
Eigenschaften, 7-6
einsetzbare MMCs, 7-7
Ziehen/Stecken, 7-18
Slave, Glossar-10
Slave-Diagnose bei Einsatz des IM 151-7 CPU als I-Slave, 6-14

SLD, A-39
SLW, A-39
Software-Uhr, 7-31
SPA, A-46
SPB, A-46
SPBB, A-46
SPBI, A-46
SPBIN, A-46
SPBN, A-46
SPBNB, A-46
Speicher
  Anwender-, Glossar-2
  Arbeits-, Glossar-2
  Backup-, Glossar-2
  Komprimieren, 7-17
  Lade-, Glossar-7
  System-, Glossar-12
Speicherbereiche
  Arbeitsspeicher, 7-13
  Ladespeicher, 7-12
  Systemspeicher, 7-13
Speicherfunktionen
  Anwenderprogramm laden, 7-15
  Hochladen von Bausteinen, 7-16
  Komprimieren, 7-17
  Laden von Bausteinen, 7-16
  Löschen von Bausteinen, 7-16
  Neustart, 7-19
  Prommen, 7-17
  RAM to ROM, 7-17
  Urlöschen, 7-19
  Warmstart, 7-19
SPL, A-47
SPM, A-47
SPMZ, A-47
SPN, A-47
SPO, A-47
SPP, A-47
SPPZ, A-47
SPS, A-47, Glossar-10
SPU, A-47
SPZ, A-47
SRD, A-39
SRW, A-39
SS, A-24
SSD, A-39
SSI, A-39
Stand-alone-Betrieb, Glossar-11
  von ET 200S, **4-5**, 4-9
Startereignis, OB, Glossar-11
Startereignisse, für OBs, 7-37
Stationsausfall, OB 86, 7-38
Stationsstatus 1 bis 3, 6-15
Statuswort, A-4
Steckplatzorientierte Adressierung der Peripheriemodule, 3-2
Steckplatzzuordnung, ET-200S, 3-2
STEP 7, Glossar-11
  Adressieroberfläche, 3-8
  Einstellungen, 4-9
  IM 151-7 CPU projektieren, 6-2
STOP
  Betriebsartenschalter, 7-4
  Betriebszustand, 7-5
  LED, 7-5
Subscriber, Glossar-11
Summenstrom, Glossar-11
SV, A-24
SYNC, Glossar-11
Systemdatenbereich, Diagnosedaten, 6-25
Systemdiagnose, 7-39, Glossar-12
Systemfunktion, B-1
  SFC, Glossar-11
Systemspeicher, 7-13, 7-20, Glossar-12
  Lokaldaten, 7-22
  Operandenbereiche, 7-20
  Prozessabbild der Ein- und Ausgänge, 7-21

T

TAD, A-41
TAK, A-41
Taktmerker, 7-39
TAR, A-31
TAR1, A-31
TAR2, A-31
TAW, A-41
TDB, A-45
Technische Daten
  allgemeine, 9-2
  des IM 151-7 CPU, 9-2
  PROFIBUS-DP, 7-2
Teilnehmer, Glossar-12
Testfunktionen, 4-9
Token, Glossar-12
Transferoperation, 3-7
TRUNC, A-42

U

U, **A-13**, A-17, A-19
U(, A-15
Übergabespeicher
  im IM 151-7 CPU, 3-5
  Zugriff im Anwenderprogramm, 3-7

UC, A-44
UD, A-18
Uhr, **7-31**, 7-39
Uhrzeitalarme, 7-39
Umrüsten, 2-2
Umrüsten , 2-11
Umverdrahten, C-4
UN, **A-13**, A-17, A-20
UN(, A-15
Urlöschen, 7-19
  CPU-interne Vorgänge, 6-5
  IM 151-7 CPU, 6-4
  mit Betriebsartenschalter, 7-4
UW, A-18

V

Montieren, 2-1, 2-3
Verdrahten, 2-1, 2-4
Verzögerung, der Ein-/Ausgänge, 8-4

W

Warmstart, 7-19
Weckalarme, 7-39
Wegweiser, durch das Handbuch, vi

X

X, **A-14**, A-17, A-19
X(, A-15
XN, **A-14**, A-17, A-20
XN(, A-15
XOD, A-18
XOW, A-18

Z

Zähler, Glossar-12
Zeiten, Glossar-12
ZR, A-25
Zugriff, auf ET 200S von PG/PC, 4-3
Zulassungen, iv
ZV, A-25
Zyklus, OB 1, 7-37
Zykluskontrollpunkt, Glossar-13
Zyklusverhalten, 7-40
Zyklusverlängerung, durch Alarme, 8-4
Zykluszeit, 8-2, Glossar-13
  Aufbau, 8-2
  verlängern, 8-3